Andreia Malucelli, Marcello Bax (Eds.)

Joint V Seminar on Ontology Research in Brazil

and VII International Workshop on Metamodels, Ontologies and Semantic Technologies

Recife, Brazil, September 19-21, 2012

Proceedings

http://ontobras-most.net84.net/

Sponsors

Organizing Institutions

Supporters

Editors' addresses:

Andreia Malucelli — malu@ppgia.pucpr.br
Programa de Pós-Graduação em Informática (PPGIa) — Pontifícia Universidade Católica do Paraná (PUCPR)
Rua Imaculada Conceição, 1155 — Prado Velho
CEP 80215-901 — Curitiba, PR — Brazil

Marcello Bax — bax@ufmg.br
Av. Fernando Ferrari, S/N, Campus Universitário de Goiabeiras, Prédio : CT-VII
Programa de Pós-Graduação em Ciência da Informação (PPGCI) — Escola de Ciência da Informação, UFMG
CEP: 31270-901 — Pampulha, BH, MG — 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 modelling 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.

This Seminar fifth edition was held simultaneously with the seventh edition of the International Workshop on Metamodels, Ontologies, Semantic Technologies (MOST), as the result of an effort of the research community in integrating the events on Ontologies which happened in Brazil in recent years. The goal was to create a unique highly scientifically qualified international forum for presenting and discussing this importante topic.

The event was organized by the Federal University of Pernambuco (UFPE), Centro de Informática (CIn). It was also supported by Sociedade Brasileira de Computação (SBC) and The International Association for Ontology and its Applications (IAOA). The event was partially funded by CAPES Foundation from the Brazilian Education Ministry, and by Foundation for the Support of Science and Technology of the Pernambuco State (FACEPE).

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 19 full-paper submissions, out of which 13 were accepted for publication and oral presentation; and 44 short-paper submissions, out of which 20 were accepted for publication and poster presentations. This volume is thus constituted by13 full papers and 20 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 2012

Andreia Malucelli
Marcello Bax

General Chairs

Fred Freitas (CIn - UFPE)
Bernadette Lóscio (CIn - UFPE)

Program Chairs

Andreia Malucelli (PUCPR)
Marcello Bax (UFMG)

Organizing Committee

Ana Carolina Salgado (CIn - UFPE)
Fabrízio Farias (CIn - UFPE)
Filipe Santana (CIn - UFPE)
Rinaldo Lima (CIn - UFPE)

Technology Committee

Luciano Cabral (CIn - UFPE)
Danusa Ribeiro (CIn - UFPE)

Steering Committee

Fernando Gauthier (UFSC)
Fred Freitas (UFPE)
Giancarlo Guizzardi (UFES)
Mara Abel (UFRGS)
Maria Claudia Cavalcanti (IME)
Maria Luiza Almeida Campos (UFF)
Maria Luiza Machado Campos (UFRJ)
Renata Vieira (PUC/RS)
Sonia Caregnato (UFRGS)

Program Committee

Alan Pedro da Silva (UFAL, Brazil)
Alcione Oliveira (UFV, Brazil)
Alexandre Rademaker (FGV, Brazil)
Alicia Diaz (UNLP, Argentina)
Ana Carolina Salgado (UFPE, Brazil)
Ana Maria de Carvalho Moura (LNCC/RJ, Brazil)
Andre Freitas (DERI, Ireland)
Andreia Malucelli (PUCPR, Brazil)
Cassia Santos (INRIA & LIG, France)
Cesar Tacla (UTFPR, Brazil)
Claudio Gutierrez (UCHILE, Chile)
Edward Hermann Haeusler (PUC-Rio, Brazil)
Emerson Paraiso (PUCPR, Brazil)
Evandro de Barros Costa (UFAL, Brazil)

Fabio Andre Porto (LNCC, Brazil)
Fernanda Baiao (UNIRIO, Brazil)
Fernanda Lima (UnB, Brazil)
Fernando Naufel do Amaral (UFF, Brazil)
Fernando Silva Parreiras (FUMEC, Brazil)
Fred Freitas (UFPE, Brazil)
Frederico Durao (C.E.S.A.R, Brazil)
Gabriela Henning (UNL, Argentine)
Gerd Wagner (BTU Cottbus, Germany)
Giancarlo Guizzardi (UFES, Brazil)
Gustavo Giménez-Lugo (UTFPR, Brazil)
Ig Ibert Bittencourt (UFAL, Brazil)
Jérôme Euzenat (INRIA & LIG, France)
José Parente de Oliveira (ITA, Brazil)
Juan Garcia-Gomez (UPV, Spain)
Laurindo Campos (INPA, Brazil)
Lucelene Lopes (PUCRS, Brazil)
Luis Márcio Spinosa (PUCPR, Brazil)
Mara Abel (UFRGS, Brazil)
Marcela Vegetti (INGAR/UTN, Argentine)
Marcelo Bax (UFMG, Brazil)
Maria Luiza Campos (UFRJ, Brazil)
Mauricio Almeida (ECI-UFMG, Brazil)
Monalessa Barcellos (UFES, Brazil)
Rafael da Rocha (UFRGS, Brazil)
Regina Braga (UFJF, Brazil)
Regina Motz (UDELAR, Uruguay)
Renata Baracho (UFMG, Brazil)
Renata Galante (UFRGS, Brazil)
Renata Vieira (PUCRS, Brazil)
Renata Wassermann (IME-USP, Brazil)
Renato Souza (FGV, Brazil)
Ricardo Falbo (UFES, Brazil)
Roberta Ferrario (LOA, Italy)
Rove Chishman (UNISINOS, Brazil)
Sandro Fiorini (UFRGS, Brazil)
Sônia Elisa Caregnato (UFRGS, Brazil)
Sean Siqueira (UNIRIO, Brazil)
Seiji Isotani (USP, Brazil)
Stefan Schulz (MUG, Austria)

Contents

Part I

Full Papers

Epistemology and medical records: an applied evaluation

Maurício B. Almeida[1], André Q. Andrade[1], Fabrício M. Mendonça[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

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

Abstract. *Medical records are crucial resources for every aspect of health-care practice. The amount and complexity of the information they bear require the use of automation. In this paper we propose a method for separating and classifying the information available in medical records, drawing on Karl Popper philosophical theories. We test this method by using descriptions of clinical cases within the scope of a biomedical project that deals with the human T cell lymphotropic virus. Our goal is to come up with a framework that allows for the organization and sharing of information in knowledge representation ontologies according to their epistemological or ontological nature.*

1. Introduction

The medical record is a complex document employed for several purposes in the healthcare realm. Proper documentation of medical encounters is one of the physician´s most important activities. Medical records have a myriad of uses in healthcare processes, such as: to support patient care, to fulfill external obligations, to support quality management [Haux, Knaup and Leiner 2007]. As a consequence of those multiple uses, medical information is a mix of facts, impressions, measurements, rules, and knowledge recording. A classification encompassing different kinds of information is required in order to represent them in systems.

There are several approaches to organizing and sharing information in medicine: information models, like HL7 [HL7 2012] and Open EHR [Garde et al. 2007]; terminologies, like MESH [Lowe and Barnett 1994]; and thesaurus, like NCI Thesaurus [NCI 2012]. An alternative that has been widely accepted for knowledge representation is the use of formal principles based on philosophical foundations. Under ideal conditions, the terms in a vocabulary would be defined free of ambiguities and overlaps in a structure called an "ontology" [Smith 2003] [Guarino 1998]. Ontologies have been widely adopted in the medical field in order to deal with the massive information produced in medicine [Rosse and Mejino 2003] [Rector and Rogers 2006].

Within the scope of the research on ontologies, a disseminated approach is the so-called "realism". In Philosophy, the term realism is widely used and controversial [MacLeod and Rubenstein 2005], but taken as a methodology, realism is extensively employed in biomedicine [Baker et al. 1999] [Grenon, Smith and Goldberg 2004], e. g, as a guiding methodology for the Open Biomedical Ontologies (OBO) Foundry [Smith et al. 2007]. Realism advocates that when scientists make claims about the types of entities that exist in reality, they are referring to entities called *universals* or *natural kinds* [Munn and Smith 2008]. Here, we call these claims "ontological information".

Some kinds of information, which are relevant for the medical field, cannot be properly represented following realistic guidelines. Examples are claims about the characteristics of signs and symptoms, which we name "epistemological information" [Bodenreider, Smith and Burgun 2004].

In previous papers [Andrade and Almeida 2011], we proposed a method for separating and classifying information available in medical records. In this paper, we extend this framework delving deeper on the theories of Karl Popper, namely the *three worlds* and *truthlikeness*, in order to create an analysis framework to be employed in the organization of the entities found in medical records. With the latter, we rely on our proposed framework to extract information from real medical records, both ontological, regarding those entities that can be represented as universals; and epistemological, which is relevant for medical practice even though it cannot be represented as universals. In addition, we rank epistemological information according to its degree of truth, with the aim of reaching a better characterization of it in ontologies.

We have been conducting the investigation that is the object of this paper, written within the scope of a biomedical project, specifically focused on human blood. The goal of this biomedical project is the development of a knowledge base for scientific and educational applications related to the human T cell lymphotropic virus (HTLV). The basis of infection by HTLV is not well-established [Verdonck et al. 2007], making the project a suitable scenario for an investigation of what could and couldn't be considered universal and to what degree a theory is close to the truth, which is carried out in the present paper.

The remainder of this paper is organized as follows: section 2 describes the theoretical basis of our investigation, presenting Popper´s theories. Section 3 contains our strategies for analyzing real data and the methodological steps taken. Section 4 presents the results of the application of our framework to real medical records. Finally, in section 5 we present a discussion and future works.

2. Background

In this section, we present the theoretical background employed as a basis for our investigation. Section 2.1 describes, briefly, the theory of three worlds of the Karl Popper and the section 2.2 his theory about truthlikeness and fallibilism.

2.1. Three worlds and medical reality

A useful approach combining reality, cognition and representations was proposed by Popper in his theory of three worlds. Popper proposes a pluralist view of the universe that recognizes at least three different but interactive worlds [Popper 1978].

According to Popper, there is a world that consists of physical bodies, such as stones, plants and animals, which is called *world 1*. World 1 can be divided into the world of *non-living physical objects* and the world of *living things* or *biological objects*. There is the mental and psychological world, called *world 2*, which includes thoughts, perceptions and observations, that is, the mental and psychological processes and subjective experiences. In world 2 we can distinguish conscious experiences from dreams, or distinguish human consciousness from animal consciousness. There is also another world, called *world 3*, which includes all content of world 2 mental processes, such as languages, scientific theories, mathematical constructions, symphonies and

sculptures. While a block of marble pertains to world 1, the creation by an artist of a sculpture using this block is a manifestation in world 3. From an ontological perspective, one can claim that world 2 and world 3 are evolutionary products of world 1.

Popper´s three worlds theory has been applied to investigations in health information science [Bawden 2002]. In the healthcare realm, world 1 consists of entities such as pains, wounds and bacteria, to mention but a few, all of them defined on the side of the patient [Ceusters and Smith 2010] [Smith et al 2006]. In world 2 one can find the cognitive representations of world 1, such as observations, interpretations and beliefs, defined both on the side of patients and physicians. World 3 is composed of concretizations of world 2 cognitive representations in diverse information artifacts, for example, terminologies, categorical systems and medical records. Moreover, diagnoses in physicians´ minds (world 2) and electronic health record entries (world 3) are related to disorders and diseases (world 1) through the relation of aboutness [Schulz and Karlsson 2011].

While Popper´s ontological view allows one to better understand the relation between entities pertaining to the world, his epistemological view proposes that every conceptualization reveals mismatches between reality and theories about reality. Though Popper's theories have been criticized [Bawden 2002], there are favorable views in which they are considered a useful model for understanding epistemological information [Abbott 2004]. Accordingly, one can find additions and improvements to Popper´s views, which propose additional sub-divisions of the original layers [Niiniluoto 1999] [Bhaskar 1978] or further subdivision of levels of reality into a material stratum, psychological stratum and social stratum [Poli 2010].

2.2. Fallibilism and truthlikeness

In addition to the three world's theory, Popper is also known for his *falsifiability criterion* and for his advocacy of *fallibilism*. According to the falsifiability criterion, scientific hypotheses are falsifiable and, therefore, scientists are able to state what empirical findings make such hypotheses false. Fallibilism is the view that no presumed knowledge, not even scientific knowledge, is absolute certain.

In this line of thought, epistemological searches are fallible. As human knowledge is incomplete, probable, and conjectural, one should seek truth but expect truthlikeness. Truthlikeness is a qualitative measure of how a theory can be more or less close to truth [Bhaskar 1978]. For example, consider these three statements in a healthcare situation: i) there are four blood groups plus a Rh factor; ii) there are four blood groups; iii) all blood has the same chemical composition. If the first assertion is true, then intuitively the second assertion has higher degree truthlikeness and approximates truth better than the third.

The medical practice is still heavily grounded in the study of signs and symptoms, which are interpreted by a physician. Medical reasoning is a sum of different cognitive practices including induction, abduction and deduction [Pottier and Planchon 2011]. In such context, in which no definitive account of truth can be reached in some cases, the notion of fallible theories being constructed from medical records is aligned with the need to search for universals.

3. Methodology

In order to develop better possibilities for medical record representation, we need to organize the kinds of information they contain. The method we propose here is composed of the following four steps.

First, we develop an analysis framework, which draws on inputs from Popper´s three worlds and we also researched by recent medical ontologies, namely, the Basic Formal Ontology (BFO) [Grenon, Smith and Goldberg 2004] as upper-level ontology to organize universals, the Ontology of General Medical Science (OGMS) [Scheuermann, Ceusters and Smith 2009] and the Information Artifact Ontology (IAO) [IAO 2012]. These ontologies were chosen because the project in which the present investigation is inserted is based on the top-level ontology BFO. It is also the ontology that provides grounds for IAO and OGMS.

As a second step, we are testing such framework on real medical records under evaluation in a biomedical project about blood diseases [Almeida, Proietti and Smith 2011]. In this paper, we will present as an example a clinical case description, which is considerably clearer than real medical records, while still requiring proper representation of the full range of medical entities. We use a generic clinical case available at Connors and Britton (2009) as a test-bed for our methodology.

In order to identify propositions within the clinical case, a domain expert transcribed the records into sentential fragments that make sense to him. The domain expert was asked to identify the reason for recording those entities and the information that is being conveyed by the representation. The transcription draws upon principles of logic and controlled languages [Fuchs et al. 2005], which allowed the identification of entities recorded in natural language, outside of the particular context in which the event took place. In addition, on the classification side, we use the rationale underpinning OGMS. This rationale is adopted to model the domain. It describes a disease as a disposition [Scheuermann, Ceusters and Smith, 2009], in which the three major stages are: etiological process, course of disease and therapeutic response. On the logical side, we took into account the fact that some parts of speech in natural language have no clear representation in logical statements.

As a third step, we consider an alternative for measuring truthlikeness, in order to classify epistemological information that came from the selected records. We took the position that epistemological information relevant in the context of medical practice cannot be registered in an ontology as a universal, following the tenets of the adopted realistic methodology [Grenon, Smith and Goldberg 2004]. It should then be registered in the form of annotations and classified according to a degree of truthlikeness. As truthlikeness is a comparative notion, we define situations which are considered true according to the current knowledge of the virus. Indeed, knowledge about the pathogenesis of infection by HTLV is fairly recent, even though this virus is endemic in several regions of the world. Genetic and immunological factors are in general the cause of the associated clinical manifestations, which may be divided into three categories: neoplastic, inflammatory and infectious [Romanelli, Caramelli and Proietti 2010]. In this step, we focus on extracting the epistemological information required to make correlations between the virus and the etiological suspects in their diverse clinical manifestations.

Finally, as fourth step, we organize the information from the medical records into four types, which are then employed in order to recommend both a data arrangement and a scenario for collaboration among different representations.

4. Results

In this section, we present the analysis framework created to organize information present in a medical record (section 4.1) and, in the section 4.2, we conduct a preliminary test of the framework by analyzing individual information entities contained in examples of the medical records.

4.1. Analysis Framework

We propose the analysis framework depicted in Fig. 1, which was created to organize information present in a medical record according to the best possibility for representation. This framework is divided into two sub-frameworks, the first one organizing the kinds of general information present in a medical record based on the three world's theory (slightly modified from Andrade and Almeida (2011) and Almeida and Andrade (2011) – a brief explanation is given for clarity); the second organizing epistemological information based on truthlikeness.

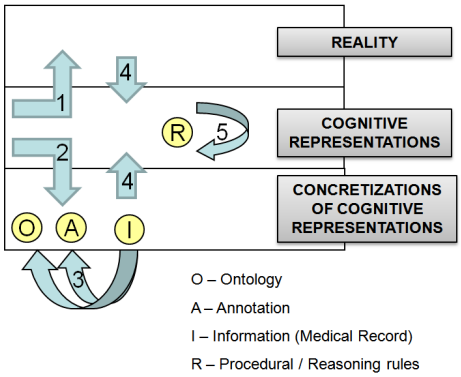


Figure 1. Framework used for analysis.

In this framework, everything begins at the level of cognitive representations when a physician observes the reality at the patient side (arrow 1). Each of these entities are filtered by cognition and represented by artifacts (arrow 2). Ontological entities (entities O) are analyzed according to strict philosophical tenets, and are based on reality itself rather than on physician´s mental representations. Examples of ontological entities are cells, anatomical features and chemical substances. These entities are directly considered in world 1 because in the realistic methodology adopted here [Grenon, Smith and Goldberg 2004] things exist in reality independently of any human beliefs. World 1 is the world of every thing that exists, observable or not. Epistemological entities are recorded in annotations (entities A). They stand for cognitive representations of reality, and may include entities without a referent in reality. Examples of these include "severity" of a pain and a sensation of "feeling well". Then, the physician creates a record (entity I) to register those representations according to their practical and theoretical knowledge (arrow 3).

Other physicians can constantly interpret records and reality (arrows 4), resulting in new cognitive representations. Finally, the physicians involved in healthcare

make judgments and process past and current information. Some of this processing of information (arrow 5) follows medical training rules, which determine the likelihood of a diagnosis and the correct interpretation of an exam result, to mention but a few. The representation of this reasoning process is also required for care continuation, which is a complementary part of the record (entity R). Examples of this include rules for interpreting lab data, as hemoglobin level < 12 g/dl means "low hemoglobin level"; and relevant negative information such as "lack of bowel alteration during episodes".

When performing this sort of analysis, we distinguish ontological information from epistemological information, the latter represented as entities in Popper´s world 3, which is equivalent in Fig. 1 to the concretizations of cognitive representations level. Within this sub-framework we recognize at least four kinds of information to be separated according to their suitability for information systems: i) information that represents aspects of reality; ii) information that represents useful constructs for the medical practice that are not empirically verifiable; iii) information that represents observations about the reality, not the reality itself; iv) information that represents observations about the physician´s understanding of the clinical situation, not about the reality.

According to the aforementioned approach, only information that represents aspects of reality can be properly represented by universals. The other three sorts identified are epistemological information. It´s worth mentioning the link between belonging to one of the three worlds and the degree of truthlikeness. The information that pertains to worlds 2 and 3 is epistemological information and it will be classified according to a degree of truthlikeness. We don´t use the notion of truthlikeness to deal with ontological information pertaining to world 1.

It is clear that (ii) and (iii) are closely related to reality, with (ii) being a surrogate for a defined state of things on the side of the patient, and (iii) an objective account of its measures. Relations that allow for proper interpretation of those statements are particular to each domain. For instance, the examination of the color of the sclera may indicate jaundice (yellow color, surrogate for liver problems) or anemia (blue color, surrogate for iron deficiency anemia). The interpretation of what such signs mean depends on training, cultural practices and subjective characteristics. There are also specific relationships with regard to lab tests, since statements like "the total bilirubin level in the blood of patient X is high" requires knowledge of the method of sampling and analysis, knowledge of the probabilistic distribution of bilirubin concentration in the normal population, consideration of measurement errors and confusion factors and understanding of the meaning of measurement units. The last category (iv) requires more attention, since medical reasoning practices include both ontological relations and ad hoc heuristic rules that are not guaranteed to hold true in the world. We consider that the information in (iv) will eventually be registered in medical records as part of a learning process.

Our proposal also includes a way of characterizing epistemological information based on its likeness to the truth. Following semantic approaches distinct from Popper´s account, such as Volpe (1995), Tichý (1978), Hilpinen (1976), we consider sentences extracted from the medical records. The semantic contents of such sentences are propositions that can be true or false.

In this sense, a simple propositional framework with three primitives (h, r, w) and the correspondent logical spaces are depicted in Fig. 2 as an example. The sentences from the associated propositional language are taken to express propositions within these logical spaces. This framework can be useful for characterizing information and scientific findings around a virus that has been studied only in the last few years, such as HTLV.

w1	h & r & w
w2	h & r & ~w
w3	h & ~r & w
w4	h & ~ r & ~w
w5	~h & r & w
w6	~h & r & ~w
w7	~h & ~r & w
w8	~h & ~r & ~w

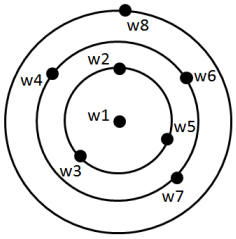


Figure 2. Three propositions generate eight levels numbered w1 to w8.

4.2. Testing the Framework

Here we conduct a preliminary test of the framework by analyzing individual information entities contained in medical records. As an example, we present a small extract of the clinical case available at Connors and Britton (2009), due to clarity and completeness of this case, and due to explicit description of reasoning processes.

After we obtain a sentential fragment from an evaluation by a domain expert, we then isolate what could be represented in realism-based ontologies following the rationale of the BFO, OGMS and IAO. After that, we arrange the information according to the sub-frameworks mentioned in section 4.1. The final results systematize the information contained in a medical record based on either their ontological or epistemological nature. To the second kind, that of an epistemological nature, we add a classification based on the level of truth.

> "A 62-year-old woman presented to the urgent care clinic with gingival bleeding after periodontal scaling of her lower-right second molar. She had undergone the procedure 5 hours before presentation, and the bleeding has persisted despite the application of pressure and ice. [...]
> The patient recalled a similar episode that had occurred 6 months earlier, also after a periodontal procedure, in which bleeding had stopped only after firm pressure had been applied and held for 6 hours. [...]
> She was otherwise in her usual state of good health. She reported no easy bruising, epistaxis, rectal bleeding, hematuria, weakness, fatigue, arthralgia, dyspnea, jaundice, abdominal pain, back pain, rash, confusion..." [Connors and Britton 2009]

In the Fig. 3, hereafter, we present samples of data obtained from the medical record in Fig. 4 and classified it according to the kinds proposed in section 3.

Data representing aspects of the reality	Data that represent useful constructs for the medical practice	Data that represents observations about the reality	Data that represents observations about the physicians understanding
Physician Woman	State of good health	Heart rate: 80 bpm	Patient class: "Emergency patient"
62 years-old	Former smoker	Blood pressure: 128/76 mmHg	Bleeding had persisted despite the application of pressure and ice
Patient report	No prior episodes of	White-cell count = 6,200	Bleeding had stopped only

	unpredictable bleeding		after firm pressure had been applied and held for 6 hours
Time of bleeding	No allergies	Lymphocytes = 37	…
Time between episodes	…	Platelet-count = 352,000	The timing of bleeding after vascular trauma is different
Aspirin	…	Creatinine = 1.4	The patients presentation suggests platelet disorder
Aspirin taken daily (rule)	…	Albumin = 3.9	**…**
Thiazide diuretic	…	Prothrombin time = 13 sec	Patient class: "Emergency patient"
Physical exam finding of that encounter	…	…	Bleeding had persisted despite the application of pressure and ice

Figure 3: Four kinds of information extracted of an example of a medical history.

This data classification was based on both the levels of representation provided in section 4.1. From the empirical assessment by physicians, the categories suggested in figure 3 were created. The relation between the proposed framework and the organization of data from medical records can be summarized as follows:

a) "Data representing aspects of reality" (column 1) were mapped from processes (1) and (2) to entities (O) (Fig. 1) - only this information that can be directly used to populate realist ontologies, since terms in ontologies refer to universals;
b) "Data that represent useful constructs for the medical practice" (column 2) were mapped from the process (1) and (2) to entities (A) (Fig. 1);
c) "Data that represents observations about the reality" (column 3) were mapped from process (3) to entities (I) (Fig. 1);
d) "Data that represents observations about the physicians understanding" (column 4) were from processes (4) mapped to entities (R) (Fig. 1).

Already the information classified in (b), (c) and (d) can to be use to support the building sets of sentence. For both, we define a set of true sentences about a blood disease following the orientation of experts. In context of the existence of the HTLV virus in a patient, a set of related sentences would be: i) HTLV cause neoplastic manifestation on human being infected by it, which we call proposition n; ii) HTLV cause inflammatory manifestation on human being infected by it, which we call proposition f; iii) HTLV cause infectious manifestation on human being infected by it, which we call proposition i. We can then consider that, in context of HTLV prevention and treatment, in a patient infected with the virus that presents both a neoplastic, an inflammatory and an infectious manifestation, those manifestations were cause by the HTLV virus. This complex situation is considered a true equivalent to the actual world we name w1. Table 1 depicts combinations of propositions ranging from w2 to w8, according to the relative closeness to the truth.

Table 1. Logical spaces for the presence of HTLV virus.

actual world = w1	**neoplasic manifestation**	**inflammatory manifestation**	**infectious manifestation**
w1	n	f	i
w2	n	f	~i
w3	n	~f	i

w4	~n	~f	~i
w5	~n	f	i
w6	~n	f	~i
w7	~n	~f	i
w8	~n	~f	~i

The truthlikeness gives us an objective criteria to evaluate the consequences of inclusion of such rules of thumb (weakness in HTLV infection is a neurologic complication) will behave in ontologies. Using this general rationale we can create "n" systems of spheres representing the situation considered real and other situations standing a logical distance from the actual world that is the truth (Fig. 4).

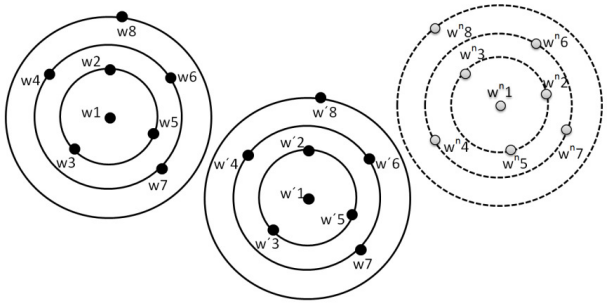


Figure 4. Logical spaces corresponding to different sets of conjunctions.

5. Discussion and future works

In this paper we presented a framework that aims to clarify the distinctions between reality, medical understanding and the recording of it, while maintaining the medical record as the main information source. Besides, we propose a way of dealing with epistemological information based on the notion of truthlikeness.

It is now well established that ontologies are an important resource to explicitly define the meaning of terms, especially when coupled with advances in description logics. Description logic is a powerful logic for describing the world, but is susceptible to inconsistencies, particularly when dealing with instance data. We advocate that realist ontologies provide a robust way of representing entities in reality, ensuring interoperability and safe inferences. This is possible because epistemological information, which can cause inconsistencies in inference processes, is not used in the ontology. Interoperabiliy is favored by the use of the top-level ontology which is the basis of the methodology adopted in this paper [Grenon, Smith and Goldberg 2004].

However, as we have shown, many entities in medical records do not have a referent in the world, being representations of epistemological evaluations by physicians and patient or measurements about real world entities. Ontologies, as pointed by Schulz in Brochhausen et al. (2011) are not "Swiss army knives for knowledge representation" and are unable to represent every single bit of knowledge required for correct interpretation of assertions. Our framework intends to make clearer which kinds of instance shouldn't be used in logical inferences, as robustness is not guaranteed. For instance, "unpredictable bleeding" is an important construct for hematologist evaluation, but a bleeding process doesn't change its way of being if someone claims it could be predicted.

Popper's theories provide a useful perspective on the different levels of reality and the relationship between theories (the ontology artifact being one of these theories) and reality *per se*. Our treatment of epistemological information seems to be an alternative to dealing with uncertainty common in the medical practice, from a logical point of view. Popper´s initiative in this regard, while essentially syntactic, entails the idea that no false theory is closer to the truth than any other. Other authors [Hilpinen 1976] [Volpe 1995] follow a semantic-oriented approach in looking for a plausible theory of distance between the semantic content of sentences. We believe that this latter approach fits well to the needs of a still-evolving subject that has to be captured in ontologies.

The proposed method has been tested in sentences obtained from real medical records, but the partial results have suggested the need for refinement. The test presented in this paper deals with a very small number of sentences and the feasibility of the approach has to be tested in more complex situations. The possibility of dealing with more complex cases is presented, for example, in Hintikka (1963). However, as a qualitative measure, truthlikeness can work as a kind of secondary metric which helps to make sense of the large amount of information in medical records.

In future works, we intend to create clear rules for dividing kinds of information in a semi-automatic fashion. It will then be possible to test our approach against a greater sample. In doing so, we aim to explore the best characteristics of different systems and which representations suitable for each sort of system.

References

Abbott, R. (2004). "Subjectivity as a concern for information science: a Popperian perspective". *Journal Information Science*, 30:95-106.

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.

Andrade, A. Q. and Almeida, M. B. (2011). "Realist representation of the medical practice: an ontological and epistemological analysis". In: *Proceedings of the 4th Ontobras;* Gramado, Brazil.

Baker, P. G.; Goble, C. A.; Bechhofer, S.; Paton, N. W.; Stevens, R. and Brass, A. (1999). "An ontology for bioinformatics applications". *Bioinformatics*, 15:510-520.

Bawden, D. (2002). "The three worlds of health information". *J. Inf. Science*, 28:51-62.

Bhaskar, R. (1978). A Realist Theory of Science. Sussex: Harvester Press.

Bodenreider, O.; Smith, B. and Burgun, A. (2004). "The Ontology-Epistemology Divide:A Case Study in Medical Terminology". In: *3rd Conference on Formal Ontology in Information Systems;* Turin, Italy. Edited by Varzi, A.; Vieu, L.

Brinkman, R. R.; Courtot, M.; Derom, D.; Fostel, J. M.; He, Y.; Lord, P.; Malone, J.; Parkinson, H.; Peters, B.; Rocca-Serra, P.; et al. (2010). "Modeling biomedical experimental processes with OBI". *Journal Biomedical Semantics*, 1 Suppl 1:S7.

Brochhausen, M.; Burgun, A.; Ceusters, W.; Hasman, A.; Leong, T. Y.; Musen, M.; Oliveira, J. L.; Peleg, M.; Rector, A. and Schulz, S. (2011). "Discussion of biomedical ontologies: toward scientific debate". *Methods Inf Med*, 50:217-236.

Ceusters, W. ; Smith, B. (2010). “Foundations for a realist ontology of mental disease”. *Journal of Biomedical Semantics*; 1:10. Url: <http://www.jbiomedsem.com/content/1/1/10>.

Connors, J. M.; Britton, K. A. (2009). “A Bloody Mystery”. *New England Journal of Medicine*; 361:e33. Url: <http://www.nejm.org/doi/full/10.1056/NEJMimc0902429>.

Fuchs, N. E.; Hofler, S.; Kaljurand, K.; Rinaldi, F. and Schneider, G. (2005). “Attempto controlled english: A knowledge representation language readable by humans and machines”. *Reasoning Web*, 3564:213-250.

Garde, S.; Hovenga, E.; Buck, J. and Knaup, P. (2007). “Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing”. *International Journal of Medical Informatics*, 76:S334-S341.

Grenon, P.; Smith, B. and Goldberg, L. (2004). “Biodynamic ontology: applying BFO in the biomedical domain”. In: *Ontologies in Medicine.* Edited by Pisanelli, D. M. Amsterdam: IOS Press; 2004: 20-38.

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

Haux, R.; Knaup, P. and Leiner, F. (2007). “On educating about medical data management - the other side of the electronic health record”. *Methods Inf Med*, 46:74-79.

Hilpinen, R. (1976). “Approximate truth and truthlikeness”. In: *Formal Methods in the Methodology of the Empirical Sciences.* Edited by Przelecki, M.; Szaniawski, A.; Wójcicki, R. Dordrecht: Reidel; 1976: 19-42.

Hintikka, J. (1963). “Distributive normal forms in first-order logic”. In: *Proceedings of the Eighth Logic Colloquium;* Amsterdam: North-Holland. Edited by Crossley, J. N.; Dummett, M. A. E. 1963: 47-90.

HL7 - Health Level Seven International [site] (2012). URL: <http://www.hl7.org/>.

IAO - Information Artifact Ontology [site] (2012). URL: <http://code.google.com/p/information-artifact-ontology/>.

Lowe, H.J. and Barnett, G.O. (1994). “Understanding and Using the Medical Subject-Headings (Mesh) Vocabulary to Perform Literature Searches”. *JAMA-J Am Med Assoc*, 271:1103-1108.

MacLeod, M. C. and Rubenstein, E. M. (2005). “Universals”. In: *Internet Encyclopedia of Philosophy*. URL: <http://www.iep.utm.edu/universa/>.

Munn, K. and Smith, B. (Eds.). (2008). “Applied Ontology. An Introduction”. Frankfurt/Paris/Lancaster/New Brunswick: Ontos, Verlag.

NCI Thesaurus - National Center Institute’s Thesaurus [site] (2012). URL: <http://ncit.nci.nih.gov>.

Niiniluoto, I. (1999). Critical scientific realism. New York: Oxford University Press.

Poli, R. (2010). “Ontology: The Categorial Stance”. In: *Theory and Applications of Ontology: Philosophical Perspectives.* 1st edition. Edited by Poli, R.; Seibt, J. Berlin: Springer; 2010: 1-22.

Popper, K. (1963). Conjectures and Refutations. New York: Routledge.

Popper, K. (1978). “Three Worlds”, In: *The tanner lecture on human values*. URL: <http://www.tannerlectures.utah.edu/lectures/documents/popper80.pdf>

Pottier, P. and Planchon, B. (2011). “Description of the mental processes occurring during clinical reasoning”. *Rev Med Interne*, 32:383-390.

Rector, A. and Rogers, J. (2006). “Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN”. *Reasoning Web*, 4126:197-231.

Rector, A. L. and Brandt, S. (2008). “Why Do It the Hard Way? The Case for an Expressive Description Logic for SNOMED”. *J Am Med Inf Assoc*, 15:744-751.

Romanelli, L. C.; Caramelli, P. and Proietti, A. B. (2010). “Human T cell lymphotropic virus (HTLV-1): when to suspect infection?”. *Rev Assoc Med Bras*, 56:340-347.

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

Scheuermann, R. H.; Ceusters, W. and Smith, B. “Toward an Ontological Treatment of Disease and Diagnosis”. In: *2009 AMIA Summit on Translational Bioinformatics;* San Francisco, CA. 2009: 116-120.

Schulz, S. and Karlsson, D. (2011). “Records and situations. Integrating contextual aspects in clinical ontologies”. In: *The 14th Annual Bio-Ontologies Meeting.* Edited by Shah, N.S. S. A.; Stephens, S.; Soldatova, L. Vienna, Austria: ISCB. 49 – 52.

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

Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L. J.; et al. (2007). “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration”. *Nature Biotechnology*, 25:1251-1255.

Smith, B.; Kusnierczyk, W.; Schober, D.; Ceusters, W. (2006). “Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain”. URL: <http://ontology.buffalo.edu/bfo/Terminology_for_Ontologies.pdf>

Tichý, P. (1978). “Verisimilitude Revisited”. *Synthese*, 38:175-196.

Verdonck, K.; Gonzalez, E.; Van Dooren, S.; Vandamme, A. M.; Vanham, G. and Gotuzzo, E. (2007). “Human T-lymphotropic virus 1: recent knowledge about an ancient infection”. *Lancet Infect Dis*, 7:266-281.

Volpe, G. (1995). “A semantic approach to comparative verisimilitude”. *The British Journal for the Philosophy of Science*, 46:563-582.

Initial approaches on Cross-Lingual Information Retrieval using Statistical Machine Translation on User Queries

Marta R. Costa-jussà, Christian Paz-Trillo and Renata Wassermann

[1] Computer Science Department
Institute of Mathematics and Statistics
University of São Paulo, Brazil
Rua do Matão 1010, São Paulo, SP 05508-090
{martarcj, cpaz, renata}@ime.usp.br

Abstract. *In this paper we propose a multilingual extension for OnAIR which is an ontology-aided information retrieval system applied to retrieve clips from a video collection. The multilingual extension basically involves allowing the user to search in several languages in a multilingual video collection. Particularly, the pair of languages we work in this paper are English and Portuguese. In order to perform query translation we use a statistical machine translation approach. Our experiments show that the multilingual system is capable of achieving almost the same quality of that obtained by the monolingual system.*

Resumo. *Neste trabalho, propomos uma extensão multilingue para OnAir que é um sistema de recuperação de informação auxiliado por uma ontologia. O sistema é usado para recuperar clips de uma coleção de vídeos. A extensão multilingue permite ao usuário fazer buscas em duas línguas em uma coleção de vídeo multilingue. Particularmente, o par de línguas que trabalhamos neste artigo são Inglês e Português. Para realizar a conversão de consulta, usamos uma abordagem estatística de tradução. As nossas experiências mostraram que o sistema multilingue é capaz de atingir quase a mesma qualidade do obtido pelo sistema monolingue.*

1. Introduction

The information society is generating a vast quantity of multilingual information. Recently, there is a growing interest in looking for information in digital videos. Generally, the user can save time, by avoiding to browse through hours of video in order to find the information he is looking for. Additionally, these videos may be in a foreign language. Although he may be able to understand the foreign language, he may not be able to formulate a query. This is the application we are focusing on in this paper in the context of the OnAIR (Ontology-Aided Information Retrieval) system. OnAIR, started in 2003, intended to allow users to look for information in video fragments through queries in natural language. The idea is save the user from the time consuming experience of having to browse through hours of video in order to find an answer for his questions.

The main contribution of this paper is the experimentation of concatenating a state-of-the-art SMT system together with an IR retrieval system that uses ontologies. This concatenation has been done for the Brazilian-Portuguese/English language pair and it can be easily be extended to other pair of languages.

The remaining of this paper is organized as follows. Next section briefly explains the related work in the area of Cross-language Information Retrieval. Section 3 describes the OnAIR structure and architecture. Then, section 4 is dedicated to the OnAIR cross-language extension. Finally, experiments and conclusions are reported in sections 5 and 6, respectively.

2. Related Work

The multilingual extension of OnAIR is basically a challenge of cross-language information retrieval (CLIR). Given a query in a source language, the aim of CLIR is retrieving related documents in a target language. (Oard and Diekema 1998) identified four types of strategies for matching a query with a set of documents in the context of CLIR by: cognate matching, document translation, query translation or interlingua techniques. From these techniques the most used are the query translation and the interlingua techniques.

Query translation methods translate user queries to the language that the documents are written. It is the most popular approach in CLIR experimental systems due to its tractability and convenience. CLIR through query translation methods has been mainly faced by using dictionary-based (i.e. using machine-readable dictionaries, MRD), machine translation (MT) and/or parallel texts techniques (Chen and Bao 2009). Among the different machine translation techniques, we have the corpus-based techniques such as statistical or example-based (Way and Gough 2005) and the rule-based techniques (Forcada 2006). In this paper we are using one of the most popular approaches nowadays which is the standard phrase-based statistical machine translation (SMT) approach (Koehn et al. 2007a).

Interlingua methods translate both documents and queries into a third representation. The approach aims at associating related textual contents among different languages by means of language-independent semantic representations. The conventional interlingua-based CLIR approach uses latent semantic indexing (LSI) for constructing a multilingual vector-space representation of a given parallel document collection (Deerwester et al. 1990; Dumais et al. 1996; Chew and Abdelali 2007). Such a representation is known to be noisy and sparse. That is why in order to obtain more efficient vector-space representations, space reduction techniques such as latent semantic indexing and probabilistic latent semantic indexing (Hofmann 1999) are applied. The new reduced-space dimensions are supposed to capture semantic relations among the words and the documents in the collection. Recent approaches have achieved interesting results by using regression canonical correlation analysis (an extension of canonical correlation analysis) where one of the dimensions is fixed and demonstrate how it can be solved efficiently (Rupnik and Shawe-Taylor 2008).

3. The OnAir system

OnAIR is in essence an information retrieval system which has been described in detail in previous studies such as (Paz-Trillo et al. 2005). In this section we briefly describe the most relevant characteristics of the system. First, we show how the information retrieval is done and, second, we show how a monolingual ontology is used for query expansion.

3.1. Information Retrieval

OnAIR relies on the vector space model (Baeza-Yates and Ribeiro-Neto 1999)for information retrieval. It was built to receive videos and keywords or their transcriptions, with timeline markers, as input, and to allow the users to query for video excerpts using natural language. When a user query is presented, OnAIR returns a list of video excerpts that best answer the user query.

The video transcriptions are pre-processed, using traditional IR techniques: stemming and stopword removal, then the vector space model is used for indexing and retrieving. As usual in traditional IR systems, some additional techniques are needed to avoid natural language difficulties like Polysemy and Synonymy.

3.2. Ontology description

Ontologies are defined in general as an explicit specification for a conceptualization (Gruber 1993). As mainly used for Information Retrieval it can be seen as a set of concepts related by hierarchies and other kind of properties in a specific domain (Ding 2001). Ontologies have been commonly used in IR through query expansion and conceptual distance measures (Paz-Trillo et al. 2005).

A domain ontology related to the topics from the videos is needed to be able to do the query expansion. By definition, query expansion is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. In particular, the domain ontology is used to measure the conceptual distance among seed query terms and new ones.

4. Cross-lingual extension

In general, a statistical machine translation system relies on the translation of a source language sentence s into a target language sentence $\hat{t}$. Among all possible target language sentences t we choose the one with the highest probability, as show in equation (1):

$$\hat{t} = \arg\max_{t} \left[P\left(t|s\right)\right] \quad (1)$$

$$= \arg\max_{t} \left[P\left(t\right) P\left(s|t\right)\right] \quad (2)$$

The probability decomposition shown in equation (2) is based on Bayes' theorem and it is known as the noisy channel approach to statistical machine translation (Brown et al. 1990). It allows to model independently the target language model $P\left(t\right)$ and the source translation model $P\left(s|t\right)$. The basic idea of this approach is to segment the given source sentence s into segments of one or more words, then each source segment is translated and the target sentence is composed from these segment translations. On the one hand, the translation model weights how likely words in the foreign language are translation of words in the source language; the language model, on the other hand, measures the fluency of hypothesis t. The search process is represented as the $\arg\max$ operation.

The translation model in the phrase-based approach (Koehn et al. 2003) is composed of phrases. A phrase is a pair of m source words and n target words extracted from

a parallel sentence that belongs to a bilingual corpus. The parallel sentences have previously been aligned at the word level (Brown et al. 1993). Then, given a parallel sentence aligned at the word level, phrases are extracted following the next criteria: we consider the words that are consecutive in both source and target sides and which are consistent with the word alignment. We consider a phrase is consistent with the word alignment if no word inside the phrase is aligned with one word outside the phrase. Finally, phrase translation probabilities are estimated as relative frequencies (Zens et al. 2002).

A language model assigns a probability to each target sentence. Standard language models are computed following the n-gram strategy, which considers sequences of n words. In order to compute the probability of an n-gram, it is assumed that the probability of observing the *ith* word in the context history of the preceding *i-1* words can be approximated by the probability of observing it in the shortened context history of the preceding *n-1* words. The main problem with this modeling is that it assigns probability zero to strings that have never seen before. One way to solve this problem is assigning non-zero probabilities to sentences they have never seen before by means of smoothing techniques (Kneser and Ney 1995).

A variation of the so-called noisy channel approach is the log-linear model (Och and Ney 2002). It allows using several models or so-called features and to weight them independently as can be seen in equation (3):

$$\hat{t} = \arg\max_{t} \left[\sum_{m=1}^{M} \lambda_m h_m(s,t) \right] \tag{3}$$

This equation should be interpreted as a maximum-entropy framework and as a generalization of equation (2) (Zens et al. 2002).

Most common additional features that are used in the maximum-entropy frameword (in addition to the standard translation and language model) are the lexical models, the word bonus and the reordering model. The lexical models are particularly useful in cases where the translation model may be sparse. For example, for phrases which may have appeared few times the translation model probability may not be well estimated. Then, the lexical models provide a probability among words (Brown et al. 1993) and they can be computed in both directions source-to-target and target-to-source. The word bonus is used to compensate the language model which benefits shorter outputs. The reordering model is used to provide reordering between phrases. For example, the lexicalized reordering model (Tillman 2004) classifies phrases by the movement they made relative to the previous used phrase, i.e., for each phrase the model learns how likely it is followed by the previous phrase (monotonous), swapped with it (swap) or not connected at all (discontinuous).

The different features or models are optimized in the decoder following the minimum error rate procedure (Och 2003). This algorithm searches for weights minimizing a given error measure, or, equivalently, maximizing a given translation metric. This algorithm enables the weights to be optimized so that the decoder produces the best translations (according to some automatic metric and one or more references) on a development set of parallel sentences.

5. Evaluation Framework

This section introduces the details of the evaluation framework. We report the translation and the information retrieval system details including corpus statistics, a description of how we built the systems and the evaluation details.

5.1. SMT data

The parallel corpus used to train the SMT system is taken from the Brazilian-Portuguese-English bilingual collections of the online issue of the scientific news Brazilian magazine REVISTA PESQUISA FAPESP (Aziz and Specia 2011). See statistics in Table 1.

		PT-BR	EN
Train	Sentences	160k	160k
	Words	4,1M	4,3M
	Vocabulary	99,5k	74.7k
Development	Sentences	1375	1375
	Words	34.3k	37.6k
	Vocabulary	6.8k	5.7k
Test	Sentences	1608	1608
	Words	36.8k	38.3k
	Vocabulary	7.3k	6.2k

Table 1. Basic characteristics of the SMT experimental dataset.

5.2. IR data

For testing the information retrieval system in Portuguese-Brazilian we used a video collection compiled from interviews with Ana Teixeira, a Brazilian artist. The interviews were made by Paula P. Braga, the domain expert and there have been used in previous studies as (Paz-Trillo et al. 2005). The interview was developed in the domain of contemporary art and the system uses a domain ontology to expand queries with related terms. To test the system, a battery of queries was synthesized both for English and Brazilian-Portuguese. Statistics of these queries and the corresponding documents for retrieving are shown in Table 2.

		PT-BR	EN
Query	Number	50	50
	Words	349	435
	Vocabulary	155	145
Documents	Number	48	-
	Words	8.2k	-
	Vocabulary	2.4k	-

Table 2. Basic characteristics of the query and documents dataset for the Ana Teixerira videos.

5.3. Translation system

In this paper, we use a system that combines the translation and the language model together with the following additional feature functions: the word and the phrase bonus and the source-to-target and target-to-source lexicon model and the reordering model. All these features have been described in section 4.

Our translation system was built using MOSES (Koehn et al. 2007b). We used the default MOSES parameters. Word alignment (built with the standard software GIZA++ (Och and Ney 2003)) was performed in both direction source-to-target and target-to-source. These word alignments were merged by using the so-called symmetrization of the *grow-diagonal-final-and* which is a sophisticated extension of the standard union operation (Koehn et al. 2005). For the translation model, we used phrases up to length 10. Phrase probability is estimated including relative frequencies in both directions (source-to-target and target-to-source), lexical weights and phrase bonus. The lexicalized reordering (Tillman 2004) is used to provide reordering accross sentences. The language model used a 5-gram with Kneser-Ney smoothing. Finally, the word bonus was used to compensate the preference of the language model for shorter outputs. All these different features were combined in equation (3) and the optimization was done using MERT software (Och 2003).

In order to evaluate the translation quality, we used BLEU (Bilingual Evaluation Understudy) (Papineni et al. 2001) which is one of the most popular SMT automatic evaluation metrics. BLEU uses a modified form of precision to compare a candidate translation against multiple reference translations. BLEU's output is a number between 0 and 1. This value indicates how similar the candidate translation and reference texts are, with values closer to 1 representing more similar texts.

We evaluated the SMT quality using in-domain and out-domain tests. The former is the one corresponding to the REVISTA PESQUISA FAPESP as shown in Table 1. The out-domain test corresponds to the queries used to test the complete CLIR system as shown in Table 2. Table 3 shows the results in terms of BLEU of the translation system when evaluated in-domain and out-domain.

Test	EN->PT-BR
In-domain	0.3649
Out-domain	0.1506

Table 3. Evaluation of the translation system in terms of BLEU.

Coherently with international evaluations such as WMT (Callison-Burch et al. 2011), the out-domain test set has a lower performance than the in-domain test set.

5.4. Comparing IR and CLIR system's performance

We performed the following experiments: two experiments using a monolingual information retrieval, recovered from previous publications (Paz-Trillo et al. 2005), and one using a cross-lingual information system. We describe the corresponding systems as follows:

1. IR system: the original system analyzed was the system described in section 3, with two configurations: *mono-keywords*, which uses only the keywords for retrieval and; *mono-kw-fulltext-05* which uses the results of retrieval using keywords and transcriptions, the best configuration for OnAIR as described in (Paz-Trillo et al. 2005)
2. CLIR system (*smt-kw-fulltext-05*): this system is the concatenation of the statistical machine translation system described in the previous section and the information retrieval system from the point above in this list.

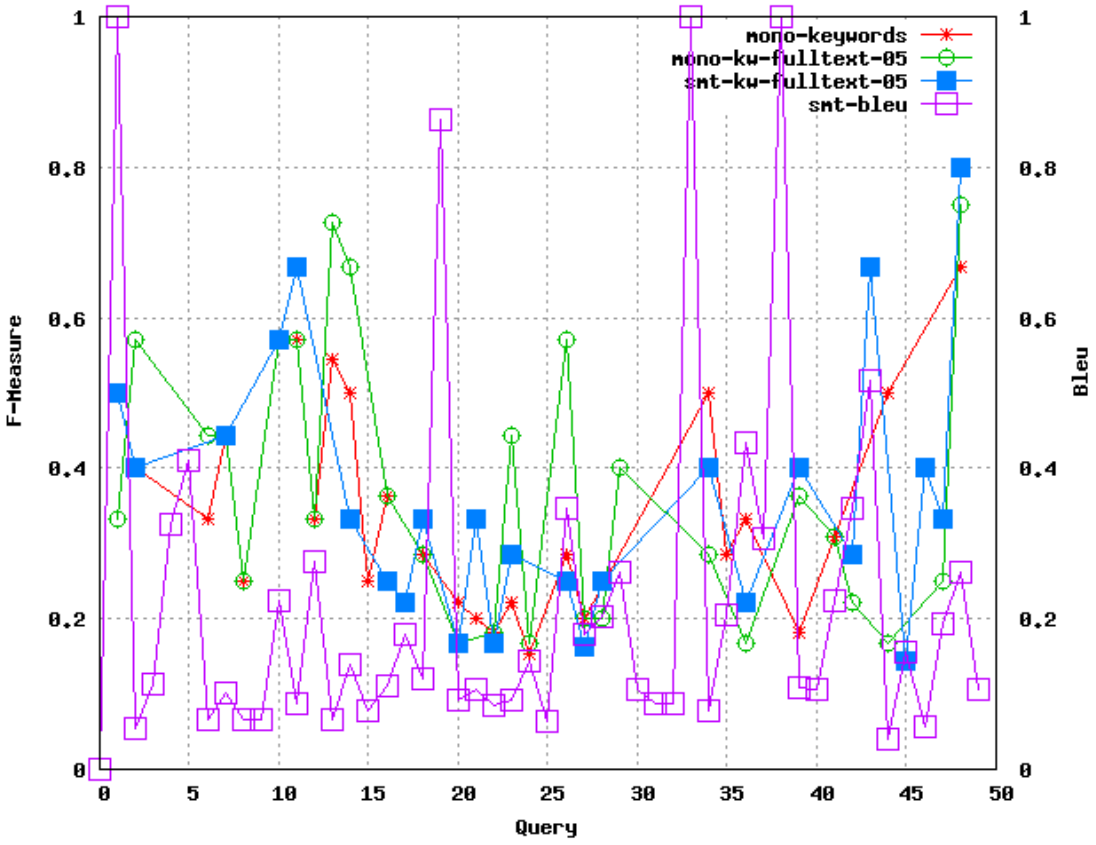


Figure 1. F-measure for the systems analyzed.

Figure 1 shows the results of the f-measure run over the 50 queries analyzed in our experiments in the three configurations presented above and the BLEU measure for the translation of each query.

Surprisingly, experiments show that the CLIR system, for specific queries, is capable of outperforming the IR system. For these queries, the translation system uses a more adequate word, which means that it would be possible to use machine translation to perform query expansion. It would be interesting to built the CLIR system with the *n*-best translations.

Figure 2 shows the f-measure in average for all systems that we experimented. Here, we observe that the f-measure of with respect to the CLIR system (*smt-kw-fulltext-05*) is slightly worst than its comparable IR system (*mono-kw-fulltext-05*). However, in

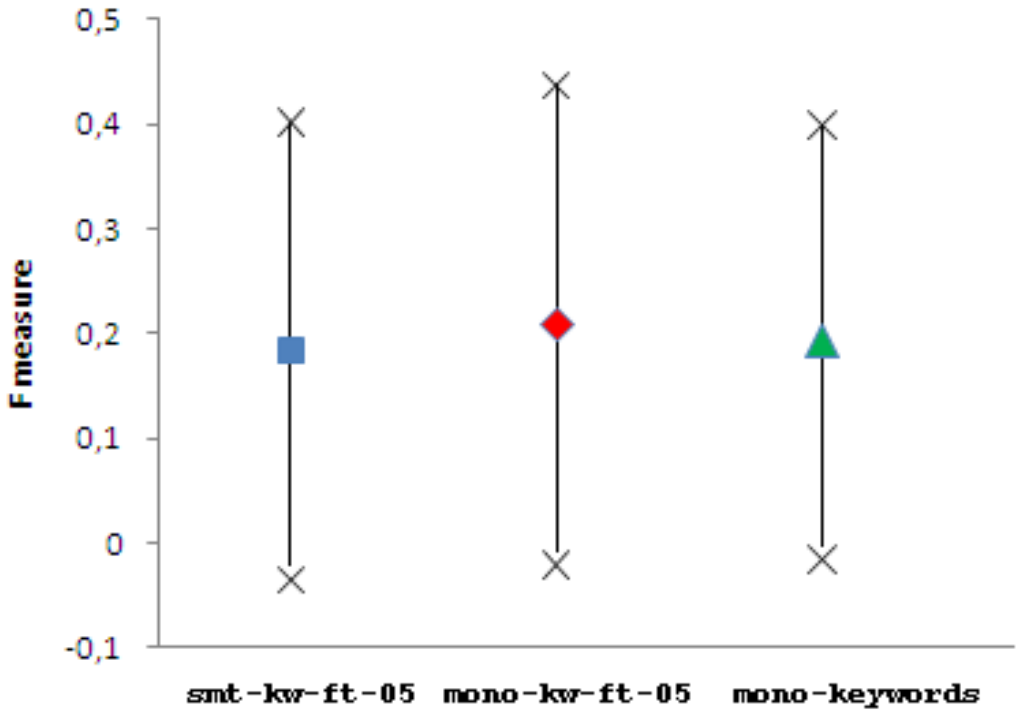


Figure 2. Average f-measure for the systems analyzed.

average, the f-measure using SMT is not highly affected when compared to the best monolingual result.

Finally, Figure 3 shows some translation examples. It shows the input to the CLIR system (*smt-kw-fulltext-05*), the corresponding translation and the corresponding reference (i.e. the input of the IR system). The two first examples report cases where the CLIR system performs worse than the IR system (*mono-kw-fulltext-05*) in terms of f-measure. The second two examples report cases where the CLIR system performs better than the IR system in terms of f-measure. Coherently, in the first case, the translation shows a poorer quality than in the second case.

6. Conclusions and future work

This paper has shown an ongoing work that generates a cross-lingual extension for the OnAIR system, which is in essence an information retrieval system using ontologies to expand queries. The cross-lingual extension has been done using a state-of-the-art statistical machine translation system. Experiments show that the best configuration for the IR system uses the results of retrieval using keywords and transcriptions. For the CLIR system, we can get competitive results using a state-of-the-art statistical machine translation system.

As further work, we want to explore different linguistic and statistical techniques (focusing on morphology and semantics) to be introduced in the state-of-the-art statistical MT system in order to correctly translate queries which are out-of-domain of the training corpus. Also it would be interesting to use MT as a query expansion method.

INPUT: How did you become an artist? TRANSLATION: Como o senhor se um artista? REFERENCE: Como você virou artista
INPUT: Do you make only interventions or also paintings, sculpture, etc? TRANSLATION: O senhor faz apenas intervenções ou também pinturas, escultura etc? REFERENCE: Você só faz intervenções ou faz também pintura, escultura, etc?
INPUT: I loved his work. TRANSLATION: Adorei seu trabalho. REFERENCE: Adorei seu trabalho.
INPUT: Have you ever exposed abroad? TRANSLATION: O senhor já exposta no exterior? REFERENCE: Você já expôs no exterior?

Figure 3. Translation examples.

7. Acknowledgements

This work has been supported by FAPESP through the OnAir project (2010/19111-9) and the *visiting researcher program* (2012/02131-2), and by the Spanish Ministry of Economy and Competitiveness through the BUCEADOR project (TEC2009-14094-C04-01) and the Juan de la Cierva fellowship program.

References

[Aziz and Specia 2011] Aziz, W. and Specia, L. (2011). Fully automatic compilation of a Portuguese-English parallel corpus for statistical machine translation. In *STIL 2011*, Cuiabá, MT.

[Baeza-Yates and Ribeiro-Neto 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley Longman.

[Brown et al. 1990] Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

[Brown et al. 1993] Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

[Callison-Burch et al. 2011] Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland.

[Chen and Bao 2009] Chen, J. and Bao, Y. (2009). Cross-language search: The case of google language tools. *First Monday*, 14(3-2).

[Chew and Abdelali 2007] Chew, P. and Abdelali, A. (2007). Benefits of the passively parallel rosetta stone? Cross-Language information retrieval with over 30 languages. In *Proc of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 45, page 872.

[Deerwester et al. 1990] Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

[Ding 2001] Ding, Y. (2001). Ir and ai: The role of ontology. In *International Conference of Asian Digital Libraries*.

[Dumais et al. 1996] Dumais, S. T., Landauer, T. K., and Littman, M. L. (1996). Automatic cross-linguistic information retrieval using latent semantic indexing. In *SIGIR96 Workshop on Cross-Linguistic Information Retrieval*.

[Forcada 2006] Forcada, M. L. (2006). Open-source machine translation: an opportunity for minor languages. In *Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages)*.

[Gruber 1993] Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220.

[Hofmann 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296.

[Kneser and Ney 1995] Kneser, R. and Ney, H. (1995). Improved backing-off for n-gram language modeling. In *IEEE Inte. Conf. on Acoustics, Speech and Signal Processing*, pages 49–52, Detroit, MI.

[Koehn et al. 2005] Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the Int. Workshop on Spoken Language Translation (IWSLT'05)*, Pittsburg, USA.

[Koehn et al. 2007a] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007a). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 177–180, Prague, Czech Republic.

[Koehn et al. 2007b] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007b). Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL*, pages 177–180, Prague, Czech Republic.

[Koehn et al. 2003] Koehn, P., Och, F., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*.

[Oard and Diekema 1998] Oard, D. W. and Diekema, A. R. (1998). Cross-Language information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 33:223–256.

[Och 2003] Och, F. (2003). Minimum Error Rate Training In Statistical Machine Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

[Och and Ney 2002] Och, F. and Ney, H. (2002). Dicriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, PA.

[Och and Ney 2003] Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Papineni et al. 2001] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: A Method for Automatic Evaluation of Machine Translation. IBM Research Report, RC22176.

[Paz-Trillo et al. 2005] Paz-Trillo, C., Wassermann, R., and Braga, P. P. (2005). An information retrieval application using ontologies. *J. Braz. Comp. Soc.*, 11(2):17–31.

[Rupnik and Shawe-Taylor 2008] Rupnik, J. and Shawe-Taylor, J. (2008). Multi-view canonical correlation analysis and cross-lingual information retrieval. In *http://videolectures.net/lms08_rupnik_rcca/*.

[Tillman 2004] Tillman, C. (2004). A Block Orientation Model for Statistical Machine Translation. In *HLT-NAACL*.

[Way and Gough 2005] Way, A. and Gough, N. (2005). Comparing example-based and statistical machine translation. *Natural Language Engineering*, 11(3):295–309.

[Zens et al. 2002] Zens, R., Och, F., and Ney, H. (2002). Phrase-based statistical machine translation. In Verlag, S., editor, *Proc. German Conference on Artificial Intelligence (KI)*.

An Operational Approach for Capturing and Tracing the Ontology Development Process

Marcela Vegetti[1], Luciana Roldán[1], Silvio Gonnet[1], Gabriela Henning[2] and Horacio Leone[1]

[1] Ingar (CONICET/UTN)
Avellaneda 3657 – S3002GJC – Santa Fe – Santa Fe – Argentina

[2] Intec (CONICET/UNL)
Güemes 3450 – S3000GLN– Santa Fe – Santa Fe – Argentina

{mvegetti,lroldan,sgonnet,hleone}@santafe-conicet.gov.ar,
ghenning@intec.unl.edu.ar

Abstract. *The history of an ontology development project, including its intermediate products, together with the executed activities, and the decisions made, might be of great importance in other future ontology developments. However, current tools supporting this kind of projects do not capture such information; thus, the process trace is lost, and any new ontology development project would start from scratch. This paper presents a framework meant to do overcome these deficiencies, allowing the capture and trace of such projects.*

1. Introduction

Until the mid-90s, ontologies were developed without addressing systematic procedures. Therefore, the ontology development process was an art rather than an engineering activity [Fernández-López et al., 1999]. In the last decade, many ontology development processes have changed from the traditional ones, performed by isolated knowledge engineers or domain experts, into collaborative processes executed by mixed teams [Bernaras et al. 1996]. In such teams, experts in knowledge acquisition and modeling, domain specialists, and experts in implementation languages collaborate to build ontologies, according to well-established methodologies. The expertise of each team member, as well as the executed activities, and the decisions made during the development process might be of great importance in future projects. However, current tools supporting ontology development processes do not capture such information; thus, the process trace is lost, and any new project would start from scratch. In fact, once a given ontology development process is finished, the things that remain are mainly isolated design products (e.g., requirement specifications, competency questions, class diagrams, specific language implementations, etc.), without an explicit representation of how these products were obtained, and with no capture of the rationale behind the process. In addition, ontology building is turning into a more professional engineering activity that needs to be managed and measured in order to obtain high quality results; and such management requires an explicit representation of the development process. The issues pointed out before constitute essential challenges that need to be addressed.

In order to tackle them, this contribution proposes ONTOTracED, a framework to represent, capture and trace ontology development processes. This paper is organized

as follows: after discussing some issues about ontology development processes in Section 2, the framework components are presented in detail in Section 3. Finally, Section 4 concludes the paper and offers paths to future work.

2. Ontology development processes

Ontology Engineering (OE) is a relatively new field concerning ontology development processes, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them. A series of methodologies have been reported in the literature in the last two decades. An extensive state-of-the-art overview of these methodologies can be found in Gómez-Pérez et al. (2004). In addition, Cristani and Cuel (2005) have proposed a framework to compare ontology engineering methodologies and evaluated the established ones accordingly. The first contributions in the field, which are due to several authors [Gruber 1993], [Grüninger and Fox 1995], [Uschold et al. 1998], [Uschold and Gruninger 1996], set the grounds for many subsequent proposals. Gruber's work [Gruber 1993] discussed some basic ontology design criteria associated with the quality of the developed ontology, as well as related to the methodology used to build it. Gruninger and Fox (1995) provided a building methodology based on Competency Questions. Methontology [Fernández-López et al. 1999] which is an ontology development process, proposed an ontology lifecycle based on evolving prototypes and specific techniques to address each activity of the approach. With emphasis on knowledge management, Staab et al. (2001) proposed On-To-Knowledge. Other approaches, related to industry or research projects, include the methods used for building CyC, SENSUS [Swartout et al. 1997] and Neon [Suárez-Figueroa et al. 2012]. These works report different principles, design criteria, and stages of the development process. However, no one is yet emerging as a clear reference [De Nicola et al. 2009]. Despite recent advances, there are few computational tools supporting the above mentioned methodologies. Neon Toolkit supports the Neon methodology and allows scheduling the stages that will be included in the design of a specific ontology. However, such tool neither captures the operations actually executed when adding a concept, a relation among concepts, etc., nor the rationale behind such operations. Consequently, there is still room for improvement in the OE field.

3. A framework to capture and trace the ontology development process

Generally, at the end of an ontology development process the things that remain are mainly unconnected design products (e.g. the requirements specification, competency questions, ontology class diagrams, the ontology implementation in a specific language, etc.), without an explicit representation of how they were obtained, and with no capture of the history and rationale behind the project. More specifically, there is no trace of the activities that have led to any of the products, the requirements imposed at each stage of the process, the actors that have performed each of the activities, and the underlying rationale behind each decision that was made. To overcome these weaknesses, this work proposes a comprehensive framework to represent, capture and trace the ontology development process, along with its associated products and their evolution.

Fig. 1 shows the main components of the proposed framework, that includes: (i) a *Conceptual Model*, which is able to represent generic design processes; (ii) an *Ontological Engineering Domain Model (OEDM)* that specifies the concepts that are

required to describe ontology development processes, and (iii) a support computational environment, named *TracOED (Tracking Ontology Engineering Designs)*, that implements both the conceptual model and the OEDM to enable the capture of specific ontology design processes, along with their associated products.

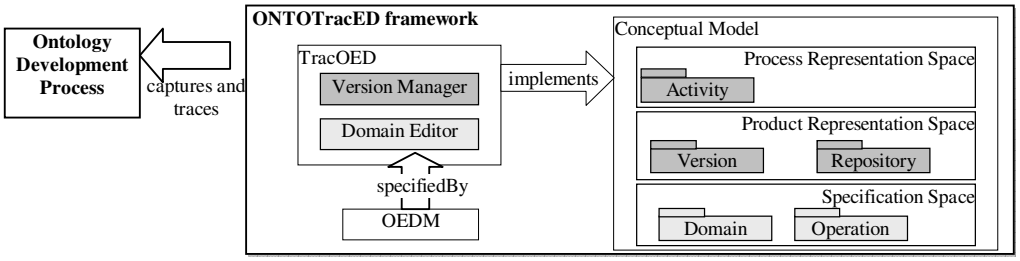


Figure 1. Components of the proposed framework.

The supporting *Conceptual Model* is based on an operational-oriented approach that envisions the ontology development project as a sequence of activities that operates on the products of the development process. The proposal defines two representation spaces to model generic design process concepts: the *Process* and *Product* spaces. In addition, a third component (the *Specification Space* in Fig. 1) is included to fully specify a flexible model that is able to represent and capture design processes pertaining to specific engineering fields.

The *Ontological Engineering Domain Model* component can represent and capture particular ontology development projects, based on building-blocks that define the products obtained, as well as the activities carried out during this type of processes. This representation includes those modeling elements that are most commonly used in the methodologies that nowadays guide ontology development processes. Among these modeling elements are: the competency question, concept, and relation concepts, etc. In order to show how this proposal may be applied when ontologists want to stick to specific methodologies and/or approaches, the ontological categories proposed by the Unified Foundational Ontology (UFO) [Guizzardi 2005] have been added to the *Ontological Engineering Domain Model*. UFO is a language to build domain ontologies that preserves the ontological commitment of the domain being modeled. It distinguishes between conceptual entities called *universals* and *individuals*. In particular, due to space limitations, this work focuses on the subsumption hierarchy of sortal universals.

TracOED is the computational environment that implements the conceptual model and incorporates the OEDM. It is based on TracED [Roldán et al. 2010], which was conceived for capturing and tracing engineering designs. The major components of TracOED are the *Domain Editor* and *Versions Manager*. By using the *Domain Editor*, the OEDM has been specified in *TracOED*. Furthermore, the editor allows this model to be further specialized, if required. On the other hand, the *Versions Manager* keeps track of the execution of a design project, as will be shown in the following sections.

3.1 Conceptual Model

The *Conceptual Model* component provides the framework foundations. This component is organized in *Process Representation*, *Product Representation* and *Specification spaces*, which are explained in this section. The *Process representation*

space models the activities being performed during an ontology development process and it is specified by the *Activity* package (Fig. 2). In particular, when tackling the development of an ontology, typical tasks are: adding concepts and relations into the ontology, defining constraints on a specific concept, analyzing whether a group of concepts, relationships and constraints satisfies a formal competency question, evaluating the ontology, deciding on alternative concepts and relations, etc. As Fig. 2 shows, such activities are represented in the model with the *BasicActivity* or *CompositeActivity* classes, depending on whether the task is atomic or it can be decomposed into a set of subactivities.

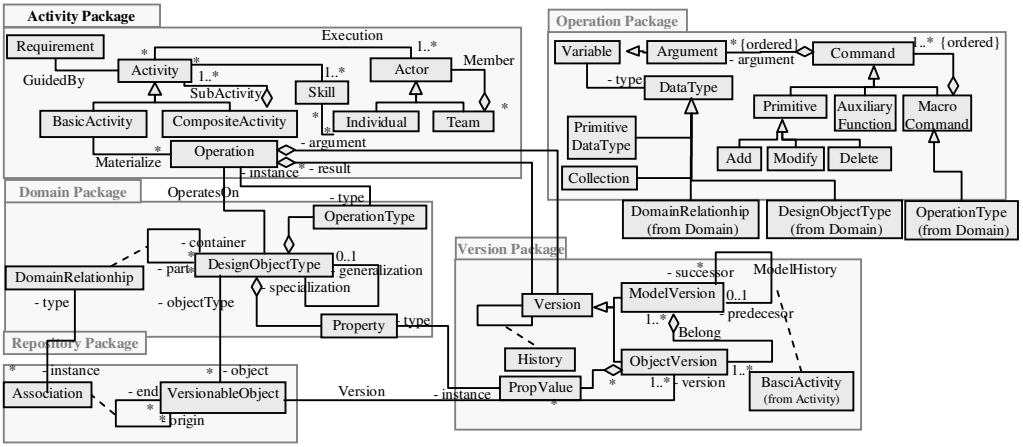


Figure 2. Conceptual Model.

In the proposed model, the execution of an activity is guided by one or more *requirements,* which specify the functional and non-functional characteristics that a development product must satisfy (e.g., in the ontology development domain, the concepts have to preserve the ontological commitment of the domain being modeled). Therefore, the ontology development process is interpreted as a series of activities led by *requirements* that are performed by *Actors.* An *Actor* may be either an *Individual* (a human being or a computational program) or a *Team.* Teams allow representing composite skills that are needed for carrying out activities. Each basic activity performed by an actor during an ontology development process is represented by the execution of a sequence of *operations*, which transforms the design objects. The operations that can be applied are domain dependent. So, it is necessary to define the allowed types of operations, as well as the modeling elements, for each specific domain.

As it was previously introduced, activities operate on the outcomes or products of the ontology design process, called *design objects* (Fig. 2). Design objects represent the various products of the development activities. Typical design objects are models of the artifact being conceived (e.g., in the ontology development domain: class diagrams, implementations in specific ontology languages, etc.), specifications to be met (i.e. competency questions, quality attributes, etc.). Design Objects may relate among themselves by domain specific relationships (*DomainRelationship* association class in Fig. 2), and can be organized in generalization-specialization hierarchies. Design objects types are described by a set of properties. Moreover, each design object type is related to a set of operation types that may be used to transform such design object.

In this proposal, the execution of an activity (materialized through a sequence of operations) transforms a design object, which thus may evolve into multiple versions. In order to represent this evolution, each *design object* is specified at two levels: the *Repository* and the *Version* packages (Fig. 2), which constitute the *Product Representation Space*. The *Repository* keeps a unique entity for each *design object* that has been created and/or modified due to the natural progress that takes place during a development project. Any entity kept in the repository is regarded as a *versionable object*. Furthermore, relationships among the different versionable objects are also maintained in the repository (*Association* class in Fig. 2). On the other hand, the *Version* level keeps the different versions resulting from the evolution of each design object, which are called *object versions*. The relationship between a *versionable object* and any of its *object versions* is captured by the *Version* association. Therefore, for a given design object, a unique instance is kept in the repository, and all the versions it assumes along the design process belong to the versions level. Fig. 2 also includes the *Design object type* class, which allows representing the various kinds of modeling elements pertaining to particular domains.

The versions package also includes the *ModelVersion* concept, which represents a set of design objects within the context in which a given design activity is carried out. Its aim is to provide a snapshot description of the state of a certain design process at a given moment. According to the proposed representation, a new *model version* m_n is generated when a *basic activity* is executed. Since each *basic activity* is *materialized* by a sequence of operations, named ϕ, the new *model version* m_n is the result of applying such sequence to the components of the previous *model version* m_p. This *predecessor model version* m_p corresponds to the context where the activity was performed and the successor one (m_n) represents the resulting context.

The *Specification Space* is defined by the *Domain* and *Operation* packages (see Fig. 2), which allow specifying the building blocks and operations of particular engineering design domains. In the context of the OntoTracED framework, this space has allowed specifying the ontological engineering domain model. The *Operation* package enables the specification of operation types and their implementations in a computational environment (TracOED in this case). This package defines the primitive operations *add*, *delete* and *modify* and also enables the specification of other operations that are applicable into the specific design domains (the ontology development domain in this work). When an *operation* is specified, it is necessary to define both its *arguments* and *body*. The *body* is comprised by some already defined commands that are available for being used in other operation specifications. They can be primitive (such as *add*, *delete*, or *modify*), *auxiliary function* commands, or previously defined operations.

3.2 Ontological Engineering Domain Model

As it was mentioned in section 3.1, the *Domain* and *Operation* packages (Specification space) of the underlying conceptual model let specify modeling elements and operations that are suitable for particular domains. This section presents the use of these packages in the specification of the *Ontological Engineering Domain Model*. Figure 3 (part a) presents a partial view of the resulting model.

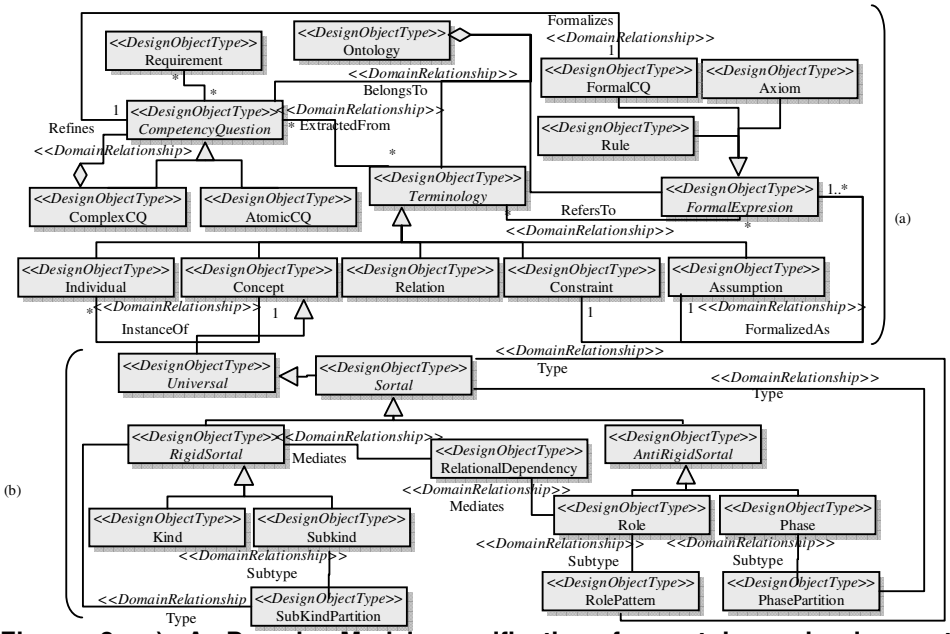


Figure 3. a) A Domain Model specification for ontology development processes. b) Design objects proposed for the development of ontologies using UFO.

There are several methodologies for building ontologies and no one is yet emerging as a clear reference. In spite of their diversity, most methodologies share structural similarities and have comparable modeling elements. In this proposal, the following components are considered to be part of the proposed domain model:

- **Competency questions** play the role of a type of requirement specification against which a given ontology can be evaluated [Gómez-Pérez et al. 2004]. They can be split off into more specific ones (*AtomicCQ* in Fig. 3), and complex competency questions (*ComplexCQ* in Fig. 3), which can be expressed in terms of simpler ones. Competency questions participate in most methodologies and they are the starting point in the identification of the ontology terminology.
- **Concepts** represent a collection of entities that share a common set of characteristics. Certain languages call them classes or frames. Concepts can be hierarchically organized by means of subsumption relationships.
- **Relations** symbolize interrelations between classes. Different languages call them properties, slots, roles, or associations.
- **Individuals** are entities that belong to a particular class. They are also called instances or members of such class.
- **Assumption** and **Constraints** represent natural language expressions that restrict the interpretation of concepts and relationships.

It is possible to distinguish between ontologies that are mainly taxonomies from the ones that model the domain in a deeper and formal way and provide more restrictions on the domain semantics. In order to represent this type of formalization it is necessary to incorporate additional design objects and operations. Therefore, the following elements have been added into the domain model:

- **Formal Competency Questions** are specification in a formal language of informal competency questions that were initially identified.
- **Axioms and rules** represent formal expressions that allow ontologists to (i) explicitly define the semantics of an ontological concept by imposing constraints on its value and/or its interactions with other concepts; (ii) verify the consistency of the knowledge represented in the ontology, and/or (iii) infer new knowledge from the explicitly stated facts.

Fig. 4 presents the functional specifications of some of the operations included in the OEDM. They give an outline of how these operations can be stated in the computational environment. From an implementation point of view, these specifications are instances of the entities defined in *Operation* Package (Fig. 2).

```
addConcept(o,cname)
   cversion:= add(cname,Concept)
   addRelationship(o,cversion, BelongsTo)
end

addInformalCQ(o,ICQname,exp)
  icqversion:= add(ICQname,AtomicCQ)
  modify(icqversion,exp)
  addRelationship(o,icqversion,BelongsTo)
end

addFormalCQ(o,exp)
  CQversion:= add(exp, FormalCQ)
  addRelationship(o,CQversion,BelongsTo)
end

toKind(o,cversion)
 n:= getname(cversion)
 kversion:=addKind(n)
 addRelationship(o,kversion,BelongsTo)
 delete(cversion)
end

addRole(o,rname,relDep,aSortal)
 rversion:= add(rname,Role)
 addRelationship(o, rversion,BelongsTo)
 rdvers:= add(relDep,RelationalDependecy)
 addRelationship(o,rdvers,BelongsTo)
 addRelationship(rversion,rdvers,Mediates)
 addRelationship(rdvers,aSortal,Mediates)
end

addSubKind(o,skname)
  skversion:=add(skname,SubKind)
  addRelationship(o,skversion,BelongsTo)
end
```

```
deriveConcept(o, cqversion,l_con)
  for each cname in l_con
       cversion:= addConcept(o,cname)
       addRelationship(cqversion,cversion,
                       ExtractedFrom)
  end for
end

formalizeCQ(ICQversion, fexp)
  o:= get(ICQversion,Ontology)
  f:= addFormalCQ(o, fexp)
  addRelationship(ICQversion,f, Formalizes)
end

applyRolePattern(o,pname,c,rname,rel,sv)
 rpversion:= add(pname,RolePatern)
 addRelationship(o,rpversion,BelongsTo)
 tversion:= type?(c)
 addRelationship(rpversion,tversion,Type)
 rversion:= addRole(o, rname,rel,sv)
 addRelationship(rpversion,rversion,Subtype)
end

applyPhasePartition(o,pname, kversion, l_con)
 ppversion:= add(pName,PhasePartition)
 addRelationship(o,ppversion,BelongsTo)
 addRelationship(ppversion,kversion,Type)
 for each cname in l_con
       phversion:= addPhase(cname)
   addRelationship(phversion,ppversion,Subtype)
end for
end
```

Figure 4. Specification of some operations belonging to the proposed model.

Fig. 4 shows some simple operations (*addConcept*, *addInformalCQ*, *addFormalCQ*) that allow adding design objects while developing a given ontology. It also shows that more complex ontological operations can as well be implemented. This is the case of the operation *formalizeCQ*, which allows formalizing a competency question, and the *deriveConcept* one. In particular, the *deriveConcept* operation allows adding into an ontology a list of new concepts that are identified from an informal competency question. The competency question object version (*cqversion*) and the list of concepts to be added (l_{con}), are the input parameters of this operation. As seen, all the proposed operations are defined in terms of primitive ones (*add*, *modify*, *delete*), auxiliary functions (*getDescription*, *getOntology*, *attachAffectedTerm*, among others), and/or operations (*addFormalCQ*, *addRelationship*).

As it was previously mentioned, the proposed OEDM defines design objects and operations to be able to handle the UFO ontological categories during the development of an ontology. Fig. 3 (part b) introduces a partial view of the resulting domain model

showing these new design objects. Table 1 presents the meanings of the concrete object types *Kind*, *SubKind*, *Phase* and *Role* and the list of applicable operations.

UFO is considered as a Pattern Language; i.e., in this language the choice of a particular design object type causes a whole pattern to be manifested [Guizzardi et al. 2011]. For example, a phase is always defined as part of a partition; a role is always played in relation to another sortal. Therefore, the adopted domain model also includes the following design patterns proposed by UFO: *SubKindPartition*, *PhasePartition* and *RolePattern* [Guizzardi et al. 2011].

Table 1. UFO Sortal Universals. Adapted from Guizzardi (2005)

UFO Ontological Categories	
Kind	A *Kind* represents rigid, relationally independent object universals that supply a principle of identity for their instances. Examples include instances of Natural Kinds (such as Person, Dog, Tree) and of artifacts (Chair, Car, Television).
SubKind	A *SubKind* is a rigid, relationally independent restriction of a substance sortal that carries the principle of identity supplied by it. An example could be the SubKind MalePerson of the Kind Person.
Phase	A *Phase* represents anti-rigid and relationally independent universals defined as part of a partition of a sortal. For instance, [Child, Teenager, Adult] is a partition of the kind Person. A Phase is always defined as part of a partition.
Role	A *Role* represents an anti-rigid and relationally dependent universal. For instance, the role student is played by an instance of the kind Person.

Proposed Operations			
Basic		**Pattern related**	
addKind	toRole	addPhasePartition	addPhase2Partition
addSubKind	toPhase	addRolePattern	addSubkind2Partition
addPhase	remKind	addSubkindPartition	remPhaseFromPartition
addRole	remSubKind	remPhasePartition	remRoleFromPartition
toKind	remPhase	remRolePattern	remSubkindFromPartition
toSubKind	remRole	remSubkindPartition	

Table 1 also presents the operations required to capture and manage the UFO-related design objects (Fig. 3 part b). It includes two groups of operations: basic ones, which comprise operations to add, delete or modify simple design objects, and pattern-related ones. These last operations are associated with the addition of the new set of design objects that follows the application of a given UFO pattern. Fig. 4 also presents the functional specification of some of these operations. As seen, the *toKind(o, cversion)* operation adds into a given ontology (*o*) a *Kind* design object (*kversion*), which is a refinement of a previously included concept (*cversion*). This operation also deletes the *cversion* concept from the current model version. Similarly, the *addRole* and *addSubKind* operations allow adding the *Role* and *SubKind* design objects to a given ontology *o*. Fig. 4 also presents the *applyRolePattern* and apply *PhasePartition* operations, which add a Role pattern and a *Phase* partition into a certain ontology, respectively. The rest of the operations are defined in a similar way by means of primitive operations (*Primitive* in Fig. 2), such as *add(skname,SubKind)*, and non-primitive ones, like *addPhase(cname)*.

3.3. TracOED

TracOED is the computational environment that implements the conceptual model and incorporates the OEDM, thus materializing the ONTOTracED framework. In order to

illustrate its features a case study is presented in this section. It is based on the development of the well known travel ontology.

As already mentioned, the *Versions Manager* enables the execution of each ontology development project, and captures its evolution based on *operations* that are accomplished and the instantiation of those *design object types* that have been specified in the Ontological Engineering Domain Model by means of the *Domain Editor* tool.

The development of the ontology starts with the definition of competency questions from which the requirements of the ontology and some initial concepts are identified. For instance, from the CQ1 competency question, which is shown below, one of the ontologists recognized the concepts *Person*, *Traveler* and *Destination*, among others.

CQ1: *Given the preferences of a traveler, the age and some constraints (economical or about the travel itself), which destinations are the most suitable?*

The identification of all the concepts from suitable competency questions marks the end of the first stage of the ontology development process. In the following stage the ontologist has to assign UFO ontological categories to the identified concepts, as well as he/she has to define new concepts falling into these categories. In this case study, the ontologist working on this part of the project considered that each of the *Person* and *Destination* concepts should be represented as a *Kind*. This decision caused the creation of a new ontology version where the *Person* and *Destination* concepts were replaced by their corresponding kinds. In addition, during this stage the ontologist gathered more domain knowledge, which allowed him/her to specify the ontology in more detail. In particular, he/she identified that a person plays the role of *Traveler* related to a *Travel Agency*. Moreover, considering the age of travelers, the involved ontologist distinguished among young, adults and old travelers. Therefore, he/she applied a phase pattern to represent this situation.

Fig. 5 presents a schema that exemplifies how the development process is captured by the *Version Manager*. The upper part of Fig. 5 shows the two ontology versions that were described above and that are inferred from the captured knowledge. In fact, the project evolves from a *Root Model Version*, which is empty, to *Model Version1* by applying the $\phi 1$ sequence of operations, which in turn is captured by the tool from the operations that were performed by the ontologist during the first stage of the process (definition of competency questions and derivation of concepts from them). Then, the evolution from *ModelVersion1* to *ModelVersion2* is caused by the operations included in $\phi 2$. These operations capture the activities carried out by the ontology developer when he/she applied the role and phase partition UFO patterns.

The first operations sequence, ϕ_1, includes the *addOntoloy, addInformalCQ* and *deriveConcept* operations that are responsible for creating the *CQ1*, *cPerson and cDestination* versionable objects at the repository level, and their first corresponding object versions ($CQ1_{v1}$, $cPerson_{v1}$ *and* $cDestination_{v1}$) at the version level. In turn, ϕ_2 comprises the *tokind, addSubKind, applyRolePattern* and *applyPhase Partition* operations. The execution of these operations has the following impact in *ModelVersion2*: (i) the addition of *kPerson* (*Kind*), (ii) the incorporation of a *RolePattern*, which comprises *kPerson*, *TravelAgency* (*SubKind)*, *rTraveler* (*Role)* and

the *clientOf* (*RelationalDependency*), (iii) the inclusion of the *byAgePh* phase partition having the *YoungTraveler*, *AdultTraveler* and *OldTraveler* phases, and (iv) the removal of the *cPerson* and *cTraveler* concepts from the current model version.

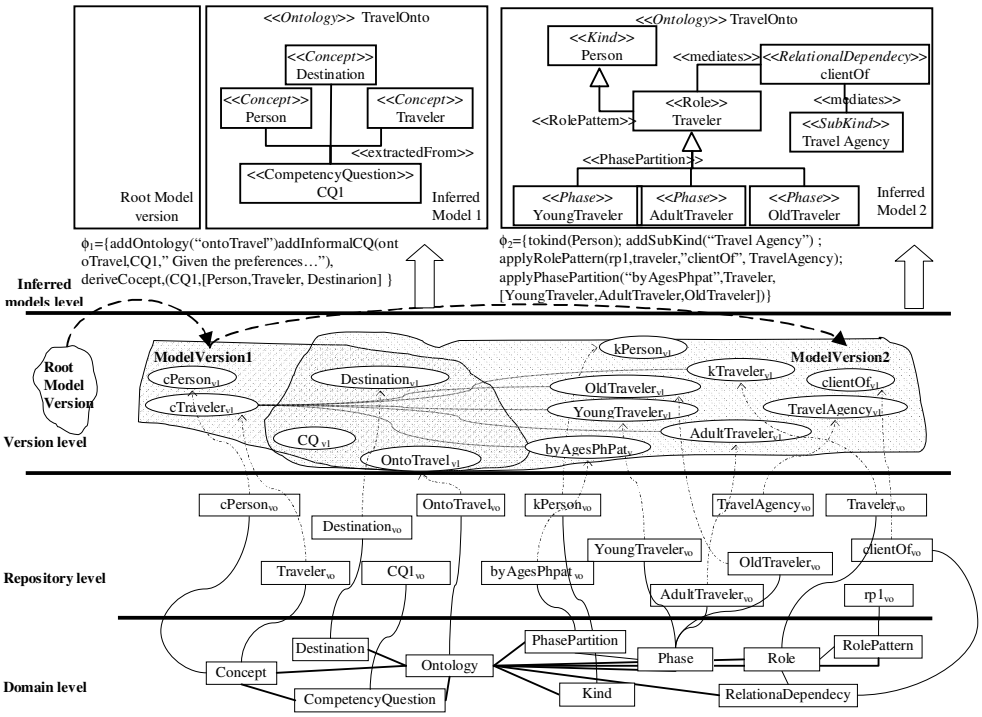


Figure 5. Specification of some operations belonging to the proposed model.

For each executed operation a version history link is created. For clarity reasons Fig. 5 only shows the version history links that relate *cTraveler* (*ObjectVersion)* in *ModelVersion1* with *kTraveler*, *OldTraveler*, *AdultTraveler*, *YoungTraveler* and *byAgesPhPart* (*ObjectVersion)* in *ModelVersion2*. By means of the history links it is possible to reconstruct the history of a given model version starting from the root one. The *Version Manager* presents such information in the so called *History Window,* which is illustrated in Fig. 6. In this pane it can be seen that TracOED allows keeping information about the development evolution of the *ontoTravel* ontology. From this knowledge it is possible to identify which are: (i) the predecessor and successors of *ModelVersion1*; (ii) the history links saving traces of the applied operation sequences, ϕ_1 and ϕ_2, which originated *ModelVersion1* and *ModelVersion2*, respectively; (iii) the set of object versions (*byAgePh*, *YoungTraveler*, *AdultTraveler* and *OldTraveler)* that appeared as a result of a given operation execution (*applyPhasePartition*).

Moreover, on the *Version Manager* History Window (Fig. 6) it is possible to see detailed data about each applied operation. For instance, this pane presents information about the time point at which a given operation was applied, who the involved actor was, and the identification of the successor object versions. In this example, the history window shows that an *applyRolePattern* operation was executed at *ModelVersion2* by mvegetti at time 11:40 -14/03/2012. It is possible to see that the execution of this operation also implied the addition of both, the *Traveler* role and the *clientOf* relational dependency.

It is important to remark that TracOED was developed with the aim of proving the proposed ideas and materializing the ONTOTracED framework. Therefore, this tool is not meant to replace traditional support environments. On the contrary, in the future TracOED should be integrated with existing ontology development tools, such as the OntoUML editor. In this way, TracOED would perform the capture of all the applied operations by working in a background mode, without being noticed by ontologists.

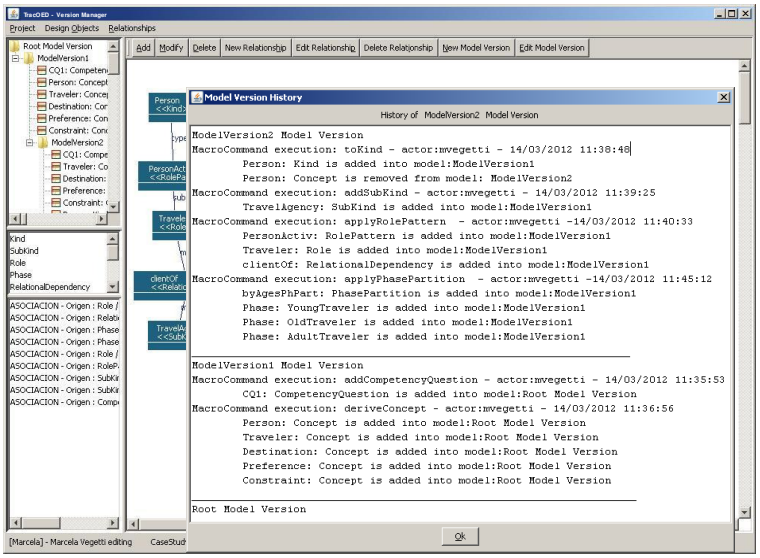


Figure 6. TracOED history window.

4. Conclusions

This contribution presents ONTOTracED, which is a framework aimed at capturing and tracing ontology development processes. The framework is based on a conceptual model of generic engineering design projects, an Ontological Engineering Domain Model, which specifies design objects and operations that are specific to ontology development processes, and a computational environment, named TracOED, which implements these models. The capabilities of TracOED have been presented and afterwards illustrated by means of a case study. The example shows that it is possible to keep track of the ontology development process along with its associated products, to store its history, allowing for the future retrieval of knowledge and experience. The proposal is flexible enough to be used in the development of ontologies that rely on particular methodologies and/or approaches, or that address particular fields. If needed, the TracOED domain editor can be used to extend the proposed Ontological Engineering Domain Model or to create a new one. To further validate the proposal, future work will be oriented to integrate TracOED with existing ontology development tools, like Protégé, the Neon Toolkit or the ontoUML editor, in such a way that its execution would take place in a background mode.

Acknowledgments

The authors wish to acknowledge the financial support received from ANPCyT (PAE-PICT-2315 and PAE-PICT-51), CONICET (PIP2754), UTN(PID 25-O117 and PID 25-0118), and UNL (CAI+D R4 N12).

References

Bernaras, A., Laresgoiti, I., Corera, J. (1996). "Building and Reusing Ontologies for Electrical Network Applications". In: the European Conference on Artificial Intelligence (ECAI'96), p. 298-302.

Cristani, M., Cuel, R. (2005). "A Survey on Ontology Creation Methodologies", *International Journal on Semantic Web and Information System*, 1, p. 49-69.

De Nicola, A., Missikoff, M., Navigli R. (2009). "A Software Engineering Approach to Ontology Building", Information Systems. 34, p. 258-275.

Fernández-López, M., Gómez-Pérez, A., Sierra, J. P. Sierra, A. P. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment, Intelligent Systems, 14, p. 37-46.

Gómez-Pérez, A., Fernandez-López, M., Corcho, O.: (2004). Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web, Springer, 2nd edition.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. PhD with Cum Laude, Telematica Instituut Fundamental Research Series, 015, Enschede, The Netherlands.

Guizzardi, G., Pinheiro das Graça, A., Guizzardi, R. (2011). "Design Patterns and Inductive Modelling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML". In: 3rd International Workshop on Ontology-Driven Information Systems (ODISE 2011).

Gruber, T.R. (1993). "A Translation Approach to Portable Ontology Specification", Knowledge Acquisition. 5, p. 199-220.

Grüninger, M, Fox, M. (1995). "Methodology for the Design and Evaluation of Ontologies." In: Skuce D (ed). IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, p. 258-269.

Roldán M. L, Gonnet S, Leone H. (2010). "TracED: A Tool for Capturing and Tracing Engineering Design Processes", Advances in Engineering Software, 41, p. 1087-1109.

Staab, S., Schnurr, H. P., Studer, R., Sure, Y. (2001)." Knowledge Process and Ontologies", IEEE Intelligent Systems. 16, p. 26-34.

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., Gangemi, A. (2012). Ontology Engineering in a Networked World, Springer, First Edition.

Swartout, B., Ramesh, P., Knight, K., Russ, T. (1997). "Toward Distributed Use of Large Scale Ontologies". In: Symposium on Ontological Engineering of AAAI .

Uschold, M., Gruninger, M. (1996). "Ontologies: Principles, Methods and Applications", Knowledge Engineering Review, 11, p- 93–155.

Uschold, M., King, M., Moralee, S., Zorgios, Y. (1998). "The Enterprise Ontology", Knowledge Engineering Review, 13, p. 31–89.

Alignment Patterns based on Unified Foundational Ontology

Natalia F. Padilha[1], Fernanda Baião[1], Kate Revoredo[1]

[1]Federal University of the State of Rio de Janeiro (UNIRIO)
– Rio de Janeiro – RJ – Brazil

{natalia.padilha,fernanda.baiao, katerevoredo}@uniriotec.br

Abstract. *Ontology alignment is the process of finding related entities in different ontologies. In this context, precise and explicit representation of conceptualizations is essential for reaching semantic integration to ensure that only data related to the same (or sufficiently similar) real-world entity are merged. Foundational ontologies describe general concepts independent of a domain and precisely define meta-properties so as to make the semantics of each concept in the ontology explicit. In this paper we show how the use of OntoUML, a conceptual modeling language based on Unified Foundational Ontology, allows the application of alignment patterns and exemplify how this approach may improve precision, recall and refine the type of the alignment.*

1. Introduction

Ontologies are explicit specifications of a conceptualization (Gruber 1995). Many domain ontologies have been developed in recent years and linking conceptualizations covering an area of common or related knowledge is a recent research problem that motivated the development of several techniques for aligning ontologies.

Ontology alignment is the process of finding related entities in different ontologies. Euzenat (2007) presents a classification of elementary alignment techniques based on the kind of data input the algorithms work on: strings (terminological), structure (structural), models (semantics) or data instances (extensional). The first two are found in the ontology descriptions. The third one requires some semantic interpretation of the ontology. The last one constitutes the actual population of an ontology.

The most difficult integration problems are caused by semantic heterogeneity (Ziegler and Dittrich 2007). Semantic integration has to ensure that only data related to the same (or sufficiently similar) real-world entity is merged. This requirement is still a challenge in the process of ontology alignment since most of the techniques discussed and implemented in automated tools so far are based on terminological or structural analyses.

On the other hand, foundational ontologies describe general concepts independent of a domain and if the domain ontologies specialize the terms introduced in a foundational ontology (Guarino 1998), it may be used as external source of common knowledge for exploiting the semantics. However, despite the benefits for building conceptual models of a domain, foundational ontologies are still insufficiently explored in the ontology alignment literature.

An essential issue for reaching semantic integration is the precision of an explicit conceptualization representation. Guizzardi (2005) addresses this issue as ontological adequacy, defined as a measure of how close a model is to the situation in reality it represents. The author presents OntoUML, a modeling language that considers the ontological distinctions and axiomatic theories put forth by the Unified Foundational Ontology (UFO) he proposes.

In this paper we show how the use of OntoUML, based on some design patterns explored in Guizzardi et al. (2011), allows the application of some alignment patterns to improve semantic integration.

Another contribution of this paper is a review in the classification of alignment approaches proposed by Euzenat (2007) concerning the technique called "Upper level, domain specific ontologies" to better organize the works that address foundational ontologies in the process of ontology alignment.

This paper is structured as follows. Section 2 presents some OntoUML design patterns that explore constraints underlying UFO. In section 3 we discuss the ontology alignment process and present a review in the classification of ontology alignment approaches. Section 4 introduces the alignment patterns based on OntoUML design patterns. In section 5 we exemplify the application of these alignment patterns. Section 6 reviews related works, followed by the conclusions in the section 7.

2. Unified Foundational Ontology

Foundational ontologies (also called upper-level or top-level ontologies) describe very general concepts, which are independent of a particular problem or domain (Guarino 1998).

UFO is one example of foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology (Guizzardi 2005). It is composed by three main parts. UFO-A is an ontology of endurants (objects). UFO-B is an ontology of perdurants (events, processes). UFO-C is an ontology of social entities (both endurants and perdurants) built on the top of UFO-A and UFO-B.

OntoUML is a conceptual modeling language designed to comply with the ontological distinctions and axiomatic theories put forth by UFO that results from a redesign process of the Unified Modeling Language (UML). The OntoUML classes, for example, make explicit the distinctions between an object and a process, types of things from their roles, among others.

A fundamental distinction in UFO is between particulars and universals. Particulars are entities that exist in reality possessing a unique identity, while universals are patterns of features, which can be realized in a number of different particulars.

UML class diagrams are intended to represent the static structure of a domain, in which classes typically represent endurant universals. The UML profile proposed by Guizzardi (2005) is a finer-grained distinction between different types of classes that represent each of the leaf ontological categories (gray entities in figure 1) specializing substantial universal types of UFO-A.

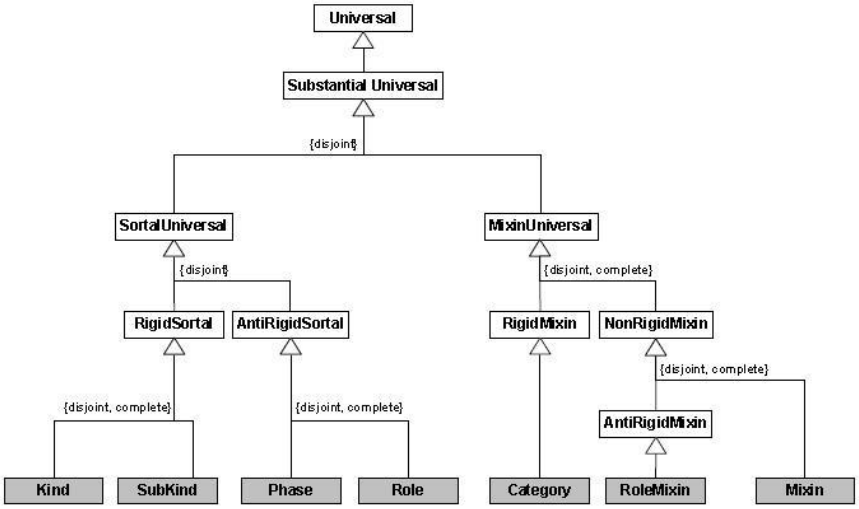


Figure 1. Ontological Distinctions in a Typology of Substantial Universals (Guizzardi 2005)

Substantials are entities that persist in time while keeping their identity (as opposed to events such as a business process or a birthday party). Constructs that represent Sortal Universals can provide a principle of identity and individuation for its instances. Mixin Universal is an abstract metaclass that represents the general properties of all mixins, i.e., non-sortals (or dispersive universals). A type is rigid iff for every instance of that type, it is necessarily an instance of that type. In contrast, a type is anti-rigid iff for every instance of the type, there is always a possible world in which it is not an instance of this type.

A kind (and subkinds) represent rigid sortals that applies necessarily to its instances, i.e., in every possible world (such as a Person, Man or Woman). A phase represents an anti-rigid sortal instantiated in a specific world or time period, but not necessarily in all of them (such as Child, Adolescent and Adult phases of a Person). A role defines an anti-rigid sortal which may be assumed in a world, but not necessarily in all possible worlds (such as a Student or a Professor role played by a Person), but once it is, this depends on its participation in a specific relation or event. Due to space restrictions, we will not define all other OntoUML categories. The design patterns presented in the next section are limited to these primitives: kind/subkind, phases and roles.

2.1. Design Patterns

The design patterns presented in this section were explored by Guizzardi et al. (2011) and are derived from the ontological foundations of OntoUML.

Subkinds can be manifested as a part of a generalization set which has as a common superclass a Kind S. In this case, the subkind classes are disjoint and complete. The Subkind Design Pattern is illustrated in figure 2(a).

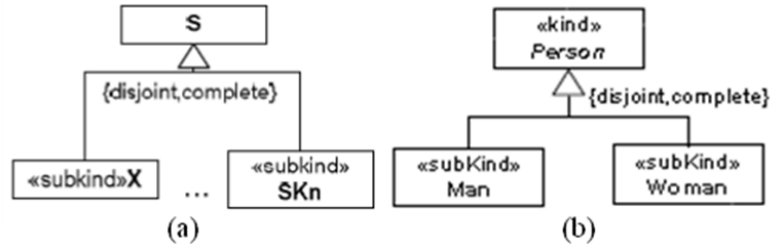


Figure 2. The Subkind Design Pattern (a) and an example of use (b) (Guizzardi et al. 2011)

Phases are always manifested as part of a phase partition (PP). In a PP there is always one unique root common supertype which is necessarily a Kind S. As well as subkinds, phases are manifested as a part of a generalization set of type S. The Phase Design Pattern is illustrated in figure 3(a).

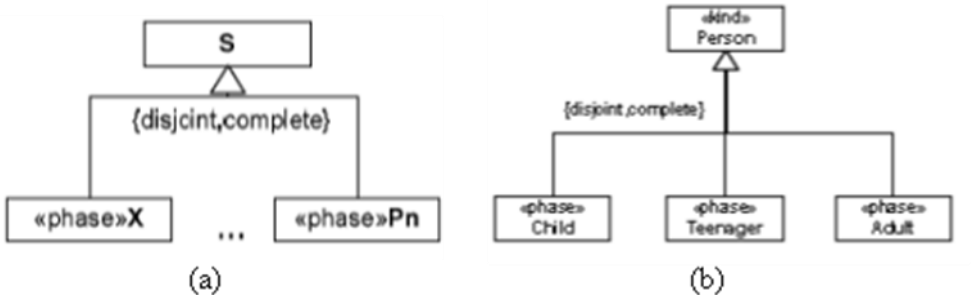


Figure 3. The Phase Design Pattern (a) and an example of use (b) (Guizzardi et al. 2011)

Roles represent (possibly successive) specializations of a Kind S by using a relational specialization condition R with another type T of the model. The Role Design Pattern is illustrated in figure 4(a).

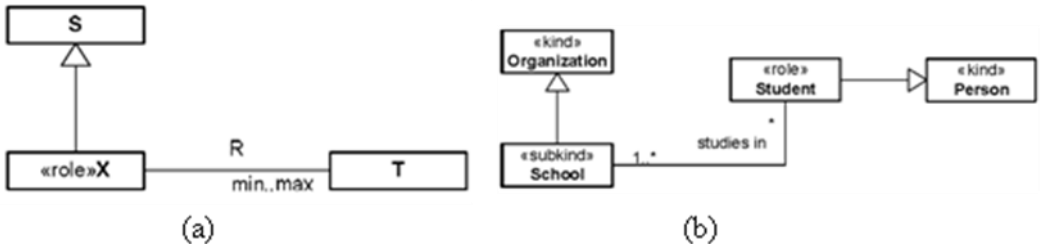


Figure 4. The Role Design Pattern (a) and an example of its use (b). Source: (Guizzardi et al. 2011)

3. Ontology Alignment

Ontology alignment is the process of finding corresponding entities (concept, relation, or instance) in two ontologies describing the same domain. A general ontology alignment function based on the vocabulary, E, of all terms e Є E, based on the set of possible ontologies, O, and based on possible alignment relations, M, is a partial function: align: E *x* O *x* O → E *x* M. Apart from one-to-one equality alignments, mostly investigated in existing work, one entity often has to be aligned not only to equal

entities, but based on another relation (e.g., subsumption). Further, there are complex composites such as a concatenation of terms (e.g., name equals first plus last name) (Ehrig 2007).

Precision and recall are commonplace measures in information retrieval and are also applied to evaluate alignment results. Precision measures the correctness of the method by the ratio of correctly found correspondences over the total number of returned correspondences. Recall is a completeness measure and considers the ratio of correctly found correspondences over the total number of expected correspondences.

3.1. Classification of ontology alignment approaches

The classification of Euzenat (2007), reproduced in figure 5, if read from the bottom up, focuses on how the techniques interpret the input information. Element-level alignment techniques compute correspondences by analyzing entities in isolation, ignoring their relations with other entities. Structure-level techniques compute correspondences by analyzing how entities appear together in a structure. Syntactic techniques interpret the input with regard to its sole structure following some clearly stated algorithm. External techniques exploit auxiliary (external) resources of a domain and common knowledge in order to interpret the input. Semantic techniques use some formal semantics to interpret the input.

If the classification is read in ascending it focus on the kinds of manipulated objects: strings (terminological), structure (structural), models (semantics) or data instances (extensional).

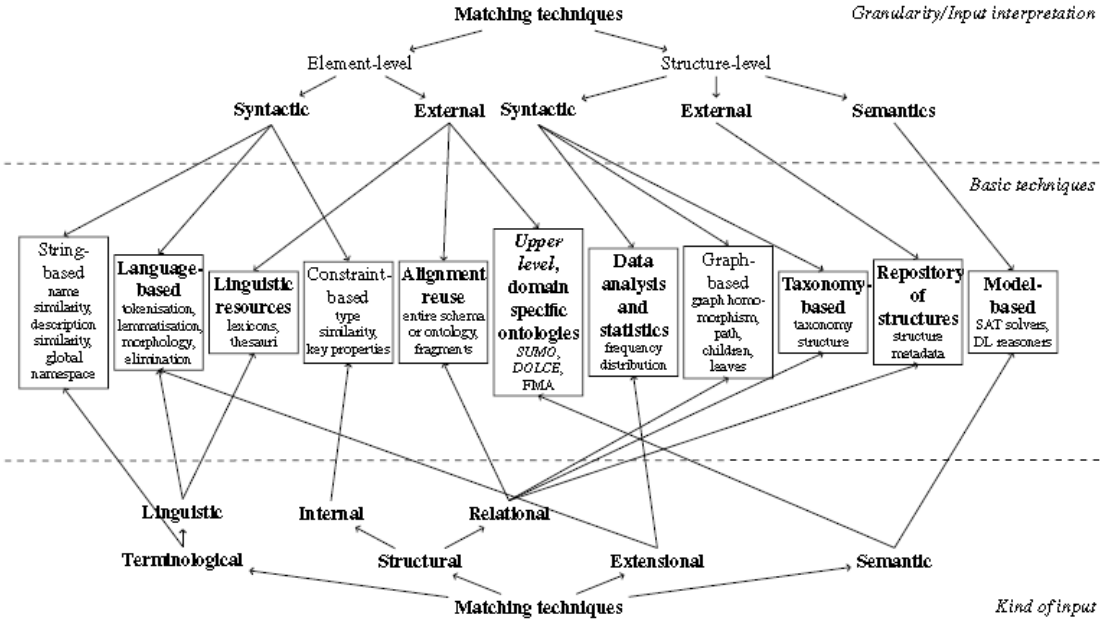


Figure 5. The retained classifications of elementary matching approaches (Euzenat 2007)

The approach proposed in this paper fits the classification described as "Upper level, domain specific ontologies", which is an element-level technique based on external semantic input. This classification groups approaches based on domain specific ontologies used as external sources of background knowledge of the particular domain

being aligned and those ones that actually exploit foundational ontologies as external sources of common knowledge. Although both cases involve the use of an external ontology, their role in the alignment process is very different. In this paper we propose a dissociation of these inputs in two techniques, as illustrated in figure 6.

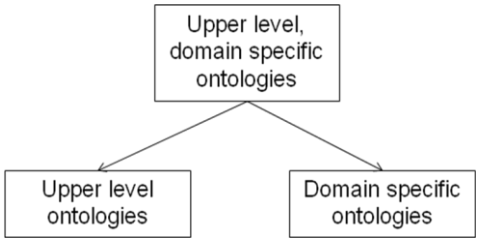


Figure 6. Dissociation of the classification "Upper level, domain specific ontologies" in "Upper level" and "Domain specific ontologies"

Considering this new classification, our approach is instantiated in the "Upper level ontologies" technique. In section 6 we will present the related work instantiated in this classification.

4. Alignment patterns based on foundational ontologies

Considering the design patterns presented in section 2 it is possible to derive some alignments patterns given below.

4.1. Subkind Alignment Pattern

This pattern consists of three rules:

Rule 1: The alignment (equivalence) of a <<kind>> class C1 of an ontology A to a <<kind>> class C2 of an ontology B is possible if all <<subkind>> classes of C1 have a corresponding class that is also a <<subkind>> of C2 in ontology B (a one-to-one equivalence is not required).

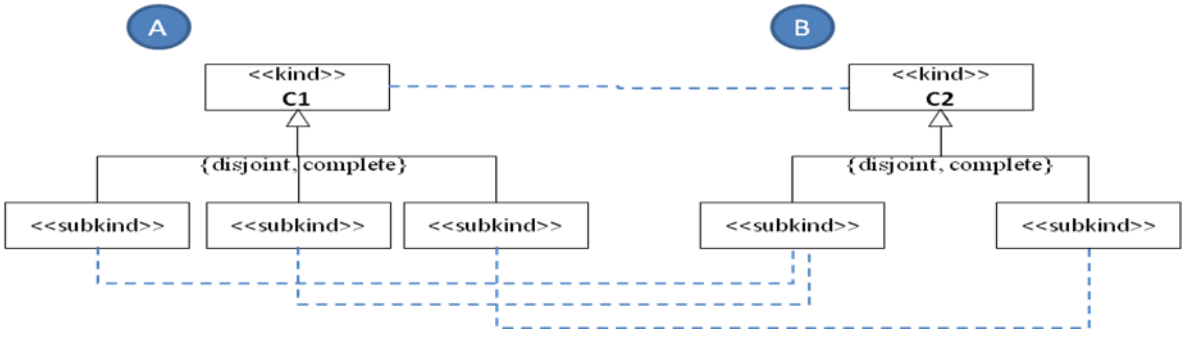


Figure 7. Rule 1 of Subkind Alignment Pattern

Rule 2: The alignment (specialization) of a <<kind>> class C2 of an ontology B to a <<kind>> class C1 of an ontology A is possible if some of the <<subkind>> classes of C1 have equivalent classes in ontology B, and these equivalences cover all the <<subkind>> classes of C2 in ontology B.

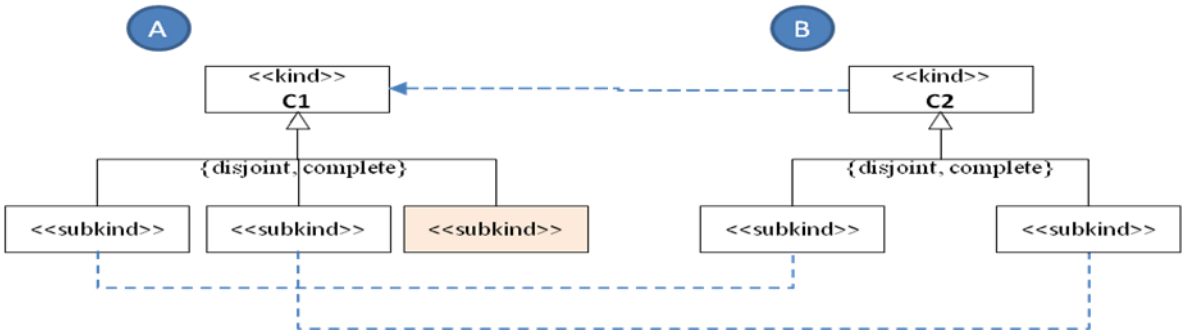


Figure 8. Rule 2 of Subkind Alignment Pattern

Rule 3: The alignment of a <<kind>> class C1 of an ontology A to a <<kind>> class C2 of an ontology B is not possible if at least one <<subkind>> class of C1 in ontology A is not aligned to a <<subkind>> class of C2 in ontology B.

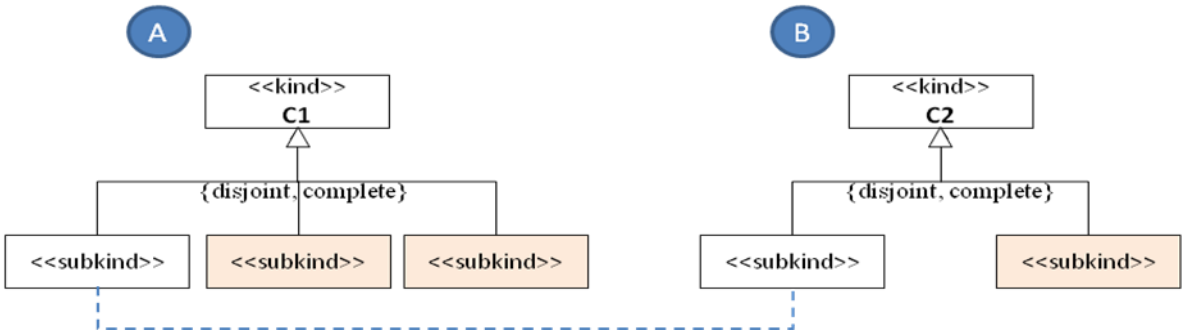


Figure 9. Rule 3 of Subkind Alignment Pattern

4.2. Phase Design Pattern

Besides the distinct semantics of Subkind and Phase classes, since both are manifested as a part of a disjoint and complete generalization set which has as a common superclass a Kind S, the rules of the Phase Design Pattern are analogous to the rules of the Subkind alignment pattern that were previously presented.

4.3. Role Design Pattern

This pattern consists of one rule:

Rule 4: The alignment (equivalence) of a <<role>> class C1 of an ontology A to a <<role>> class C2 of an ontology B is only possible if the <<kind>> rigid class that the <<role>> classes specialize, and the relation its instantiation depends on, are aligned to each other.

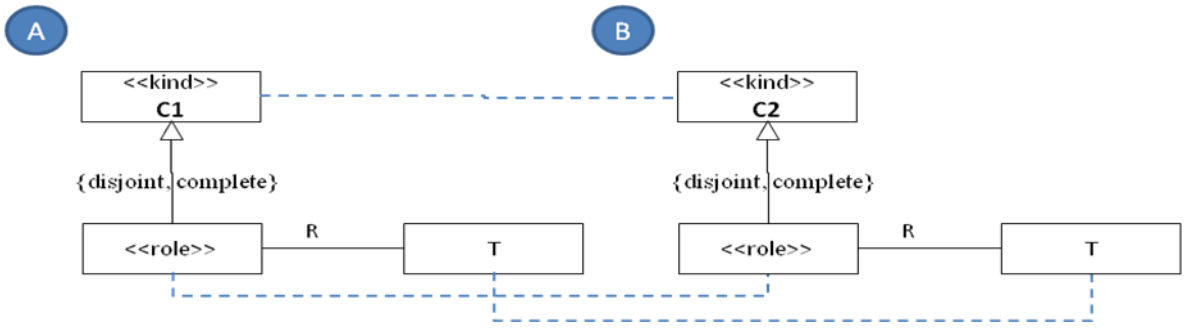


Figure 10. Rule 4 of Role Design Pattern

5. Examples of use

Is this section we will exemplify the application of the alignment patterns presented in section 4. The ontologies describe the domain of organizing conferences, which corresponds to one track of the Ontology Alignment Evaluation Initiative (OAEI) 2011[1]. The ontologies and the reference alignments indicated by the initiative are available on the Conference Track[2].

A prerequisite for application of the alignment patterns is that the ontologies to be aligned must comply with the UFO constraints set out by Guizzardi (2005) and it is necessary the identification of the OntoUML stereotype applicable to each class, considering the design patterns presented in section 2. Because the conference domain is well understandable for every researcher, this task was executed by the authors for the fragments discussed in this paper.

We have analyzed the alignment of two ontologies identified in Table 1 by considering the submitted alignments results of tools evaluated in group 1, which consists of best evaluated matchers of the track. We will explore two common errors committed by the four matchers of this group, one affecting the precision and other the recall measure.

Table 1. Ontologies Iasted and SigKdd

Name	Type	Number of classes
Iasted	Web*	140
SigKdd	Web*	49
* Ontologies have been based upon actual conference (series) and its web pages		

All four matchers have identified a correspondence between the classes Iasted::Document and SigKdd::Document that is not indicated by the reference alignment (which harms the precision). The fragments are illustrated in figure11.

In this case, both Document classes are of the type kind and correspond to a generalization set of subkind classes, disjoint and complete.

[1] http://oaei.ontologymatching.org/

[2] http://oaei.ontologymatching.org/2011/conference/index.html

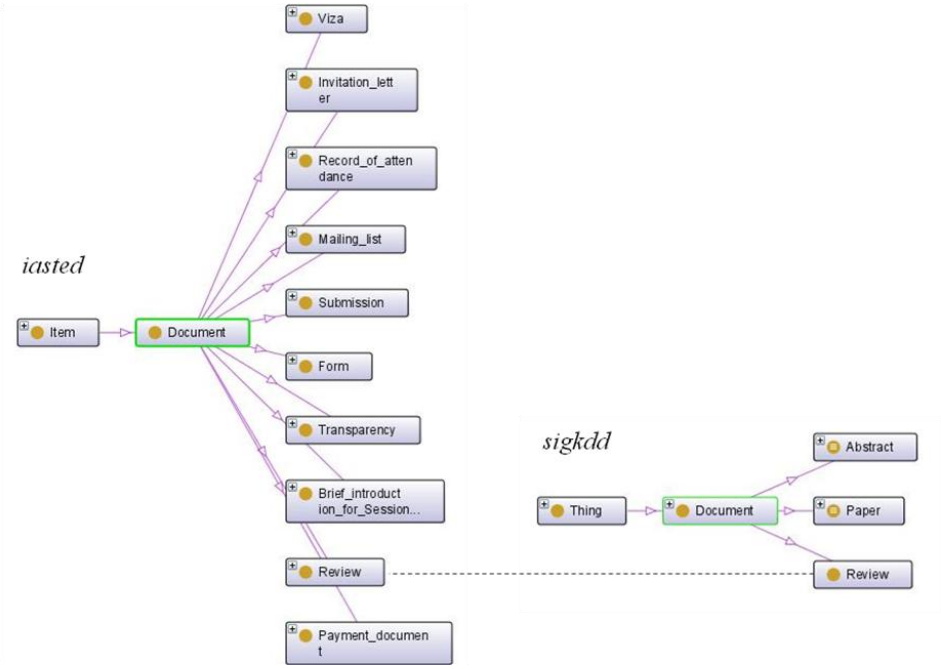


Figure 11. Alignment between Iasted::Document and SigKdd::Document

Dotted lines indicate the expected alignment between the classes of these ontologies fragments. The set of matchers considered have identified the correspondence between Iasted::Review and SigKdd::Review, besides the wrong correspondence between the Document classes. In this context, the application of Rule 3 would reject the correspondence between Iasted::Document and SigKdd::Document since some of their subkind classes are not aligned.

The other common error is that all four matchers could not identify the reference alignment between Iasted::Sponsor and SigKdd::Sponzor indicated by the reference alignment (thus reducing recall). The fragments are illustrated in figure 12.

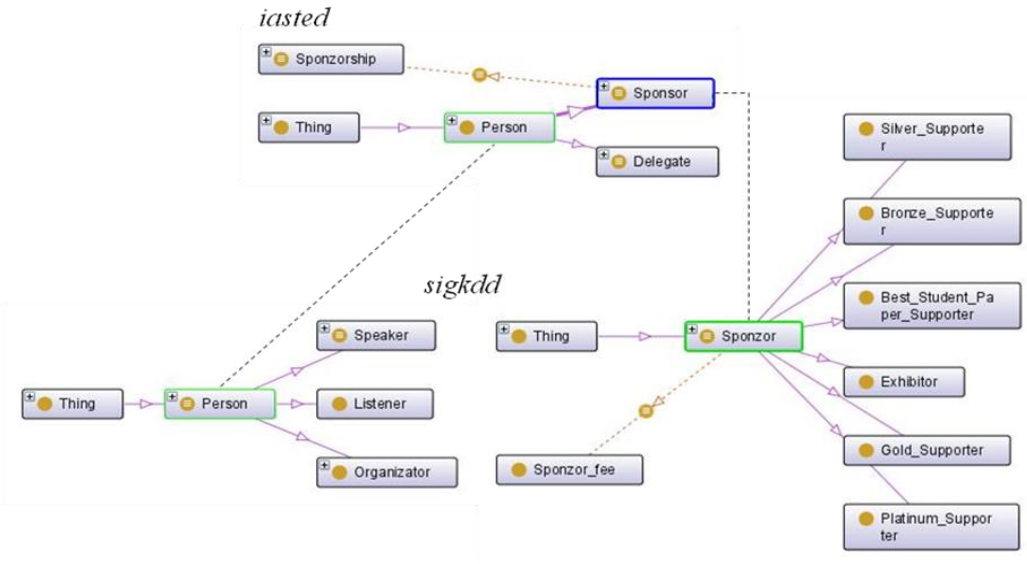


Figure 12. Alignment between Iasted::Sponsor and SigKdd::Sponzor

In this example, both Sponsor and Sponzor classes are stereotyped as roles. In Iasted ontology a Sponsor is a role played by a person that gives some Sponzorship. In SigKdd ontology a Sponzor is a role characterized by the payment of a Sponzor_fee. However, the kind class specialized by this role is not explicit in this ontology. Based on Role Design Pattern, we define this role as a specialization of the class Person, already defined in this ontology and aligned to the class Iasted::Person. With this redesign, the rigid kind classes that these roles specialize are aligned to each other, which itself brings additional information to allow the identification of the alignment between Sponsor and Sponzor classes. However, by Rule 4, to guarantee the alignment between Iasted::Sponsor and SigKdd::Sponzor, Iasted::Sponzorship and SigKdd::Sponzor_fee must be aligned, which suggests an update in the reference alignment to include this equivalence. Otherwise, this correspondence would be rejected.

In figure 13 there are fragments of other two ontologies of the same conference domain, identified in Table 2.

Table 2. Ontologies Edas and Cmt

Name	Type	Number of classes
Edas	Tool*	104
Cmt	Tool*	36
* Ontologies have been based upon actual software tool for conference organisation support		

The question identified here is not an error that affects the precision or recall, but a review of the type of the correspondence identified.

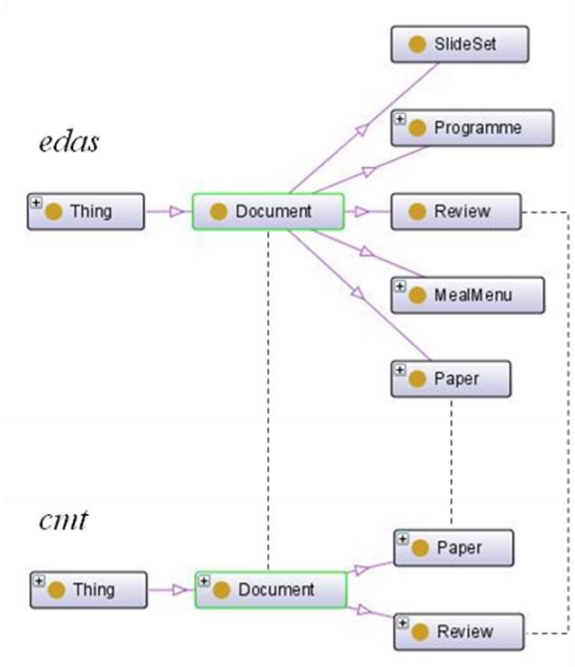


Figure 13. Alignment between Edas::Document and Cmt::Document

In this case, both Document classes are of the type kind and correspond to a generalization set of disjoint and complete subkind classes, as the first example. The

dotted lines indicate the reference alignments that mean equivalence relation. Since part of <<subkind>> classes of Edas ontology have equivalent classes in Cmt ontology, and these equivalences cover all the <<subkind>> classes of Cmt ontology, the alignment between the classes Document could be refined considering that Cmt::Document is a specialization of Edas::Document.

6. Related Work

One main point that has guided the development of the approach presented in Silva et al. (2011) is the use of foundational ontologies. To establish the relationship among the foundational ontology and the domain ontologies, for each first-level concept at the domain ontology, a foundational concept was associated. Thus, the result is a unique integrated ontology, composed by the domain ontology and some of the meta-categories of a foundational ontology. Despite considering foundational ontologies, the additional information they provide was relevant for the taxonomic similarity measure (structural input) implemented by the matcher used for the tests, as it becomes possible to compare upper-level concepts in the hierarchy when a candidate pair of concepts is under analysis.

The approach presented in this paper, in turn, proposes a directly use of foundational ontologies to improve semantic integration by considering some alignment patterns based on meta-properties of the OntoUML constructs, with the determination of rules to be applied during the alignment process. Considering the suggested dissociation of the “Upper level” and “Domain specific ontologies” it is instantiated in the “Upper level ontologies” technique.

Other works address foundational ontologies in the context of ontology alignment but they are more directly related to the use of domain ontologies to support the alignment of other ontologies on the same domain. In Mascardi et al. (2010) the techniques applied to associate the classes of the domain ontologies to the classes of the foundational ontologies are typically used to associate concepts of domain ontologies. A higher precision was only obtained with foundational ontologies that include many domain-specific concepts in addition to the upper-level ones. In Gonçalves et al. (2011) the hypothesis is that a domain reference ontology that considers the ontological distinctions of OntoUML can be employed to achieve semantic integration between data standards. The hypothesis is tested by means of an experiment that uses an electrocardiogram (ECG) ontology and conceptual models of the ECG standards. Considering the suggested dissociation of the “Upper level” and “Domain specific ontologies”, these approaches would be instantiated in the “Domain specific ontologies”.

7. Conclusion and Future Work

Ontology alignment is an active research area and some challenges consider semantic issues. In this paper we discussed how the use of OntoUML oriented by some design patterns improves the ontological adequacy of the ontologies being aligned and allows the application of some rules based on alignment patterns. We have used some ontologies from the main initiative for evaluation of ontology alignment to demonstrate how the design patterns and the alignment patterns may improve precision, recall and

refine the type of the alignment, with a manually performed example. However, the process of annotation the classes with the correct OntoUML stereotypes can be assisted by a software tool (Benevides and Guizzardi 2009).

Another contribution of the paper is a review in the classification of Euzenat (2007) concerning the technique called "Upper level, domain specific ontologies" to better organize the works that address foundational ontologies in the process of ontology alignment.

Future work includes formalization of indicative and restrictive rules based on the meta-properties of a larger set of constructs of OntoUML to be applied during the alignment process. Moreover, the automatization of the proposal and its application in complete scenarios will also be considered.

References

Benevides, A. B., Guizzardi, G. (2009) "A Model-Based Tool for Conceptual Modeling and Domain Ontology Engineering in OntoUML", In: ICEIS 2009, pp. 528-538

Ehrig, M. (2007), Ontology Alignment: Bridging the Semantic Gap, Springer

Euzenat, J. and Shvaiko, P. (2007), Ontology Matching, Springer

Gonçalves, B., Guizzardi, G. and Pereira Filho, J. G. (2011) "Using an ECG reference ontology for semantic interoperability of ECG data", In: Journal of Biomedical Informatics, vol. 44 , pp 126–136

Gruber, T. R. (1995) "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", In: International Journal of Human and Computer Studies, vol. 43, issues 5/6. pp. 907–928

Guarino, N. (1998), Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction and Integration

Guizzardi, G., Graças, A. P and Guizzardi, R. S. S. (2011) "Design Patterns and Inductive Modelling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML", In: 3rd International Workshop on Ontology-Driven Information Systems (ODISE 2011), London, UK

Guizzardi, G. (2005) Ontological Foundations for Structural Conceptual Models, Ph.D. Thesis, University of Twente, The Netherlands

Mascardi, V., Locoro, A. and Rosso, P. (2010) "Automatic Ontology Matching Via Upper Ontologies: A Systematic Evaluation" In: IEEE Transactions on Knowledge and Data Engineering, vol. 22, n. 5, pp. 609–623

Silva, V.S., Campos, M. L. M, Silva, J. C. P. and Cavalcanti, M. C. (2011) "An Approach for the Alignment of Biomedical Ontologies based on Foundational Ontologies", In: Journal of Information and Data Management, vol. 2, n. 3, pp. 557–572

Ziegler, P. and Dittrich, K. R. (2007) "Data Integration - Problems, Approaches, and Perspectives", In: Krogstie, J., Opdahl, A. L. and Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 39–58. Springer, Heidelberg

Modelling Geometric Objects with ISO 15926: Three proposals with a comparative analysis

Geiza M. Hamazaki da Silva[1,2], **Bruno Lopes**[3], **Gabriel B. Monteiro Lopes**[2]

[1] Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur, 296, Urca – Rio de Janeiro-RJ – Brazil

[2]Computer Graphics Technology Laboratory
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro-RJ - Brazil

[3]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro-RJ - Brazil

Abstract. *In the field of Oil & Gas, the ISO15926 proposes a standard for integration, sharing, exchange and delivery of data between computer systems based on the standardization of data formats and an ontology approach to represent common industry classes and relations. Due to the structure and the large number of terms defined at this standard, the complexity of creation information models is high. This aims to consolidate a methodology for modelling geometric objects following the structure of ISO 15926. We take into account the need for complete abstraction between geometry and business data. Three approaches are presented with a comparative analysis, which should reveal the appropriate practice to be adopted both in manual and in software supported ISO15926 compliant information modeling.*

1. Introduction

The National Institute of Standards and Technology (NIST) reported in 2004 that the costs generated by lack of proper interoperability between systems used in capital projects in the United States were around US$15.8 billion per year. This was in great part due to the absence of international standards for interoperability used in the projects that were analyzed by NIST. The use the information technologies to effectively integrate design, construction and business processes was not widely observed [Lee et al. 2012, Gallaher et al. 2004].

Aiming to promote interoperability for industrial automation systems for process plants, the ISO 15926 standard is designed to simplify the integration of data to support the life-cycle activities and processes of production facilities (as in Fig.1). The ISO 15926 standard is concerned with the storage of information, constructing knowledge bases for integration of life-cycle data for process plants including oil and gas production facilities [ISO 15926-1 2004]. These knowledge bases are modeled with structures in "First-Order Logic" and implemented based on an ontology approach to information, consistent with the W3C Web Ontology Language (OWL) [d'Aquin and Noy 2012, Consortium 2004].

The OWL standard is used to represent common industry terms that are mapped to the ontology with classes and relationships [Batres et al. 2007]. These terms are modeled

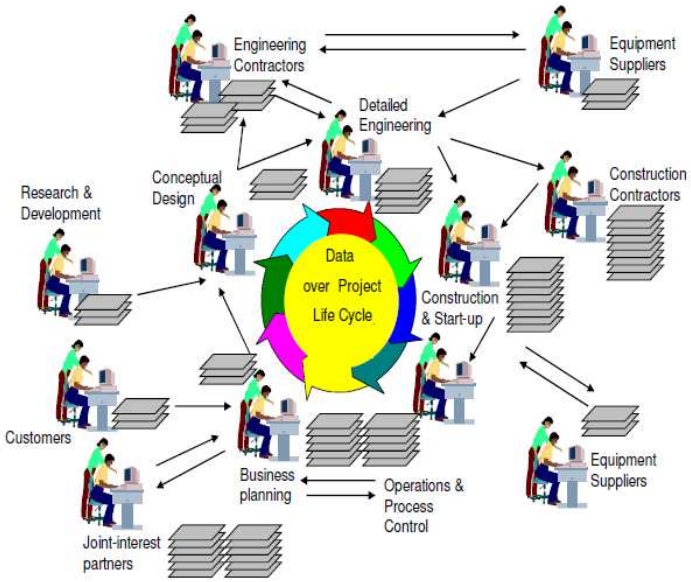


Figure 1. Data over Project Life Cycle – modified from [Pawsey 2010]

using the data model [ISO 15926-2 2003] and the initial reference data set (ISO 15926-4, 2007) which are shared databases or data warehouses used to describe industrial project lifecycle concepts. The ISO 15926 standard consists of several parts. Some of them are published, such as parts 1, 2, 3, 4, 7 and 8, while others are under development (see [IRING User Group 2012]).

Due to, among others, the high complexity of modelling concepts by the structure of the standard and the large amount of terms defined by its ontology, there is no consolidated methodology for information modelling in ISO 15926. It is essential for the usability of this standard that this complexity be hidden by the template use [ISO 15926-7 2011]. The most basic templates must be modelled using only entities from ISO 15926 - Part2 and ISO 15926 - Part4 as a requirement to be compatible with its conceptual model. Therefore, this basic model may be specialized to accommodate any field of engineering knowledge, as geometry, whose relevance is present in any Engineering schematic, 3D model, datasheets etc.

Engineers working on Capital projects use Computer-Aided Design and Drafting systems (CADD), which, for representing 3D and 2D schematics, ultimately use geometric objects (or primitives), such as: ellipses, polylines etc. Thus, to interoperate geometry related information, a standard is required for the structured data that describes the geometric objects. This is offered by the ISO 15926 Part3 [ISO 15926-3 2007], which defines the catalog of geometry and topology terms.

This work presents three approaches for modeling geometric objects following ISO 15926 and a comparative analysis among them. In the next session we present a brief introduction to this standard and after that we present our proposals. Then, we present the conclusions, work in progress and future work.

2. ISO 15926

The ISO 15926 standard (Industrial automation systems and integration, integration of life-cycle data for process plants including oil and gas production facilities [ISO 15926-1 2004]) consists of several parts. Some of them are published, like parts 1, 2, 3, 4, 7 and 8. At the time of this publication, parts 7 and 8 of ISO 15926 had been submitted to the ISO standard approval process, under TC184/SC 4. What follows is a brief introduction to the published ISO 15926 parts.

Part 1: Overview and fundamental principles [ISO 15926-1 2004] – Specifies a representation of information associated with engineering, construction and operation of process plants.
Part 2: Data Model [ISO 15926-2 2003] – Describes the entities used by the standard to represent the process plant life-cycle information. It is designed to be used in conjunction with reference data [ISO 15926-4 2007]: default instances that represent information common to users and process plants.
Part 3: Geometry and Topology [ISO 15926-3 2007] – Defines objects in the reference to data library for geometry and topology. It is based on ISO 10303 [ISO 10303-1 1994] and the dictionary of standard shapes are extracted from the ISO 10303-42 [ISO 10303-42 2003] and ISO 10303-104 [ISO 10303-104 2000].
Part 4: Reference Data Library [ISO 15926-4 2007] – Support for a specific life cycle depends on the use of appropriate reference data based on the data model [ISO 15926-2 2003].
Part 6: It defines a methodology for development and validation of reference data.
Part 7: Templates Implementation methods for the integration of distributed systems [ISO 15926-7 2011]. A template is seen as a data schema and the part 7 describes a catalog of templates and defines an implementation-independent template methodology for definition, verification, expansion of templates, as well as presenting an initial set of templates to allow the use of the conceptual model ISO 15926- Part2. It consists of the definition of the signature and axioms in first-order logic; verification and expansion are done with the software Template Expander.
Part 8: Implementation methods for the integration of distributed systems – OWL implementation. This part defines the specification for data exchange and lifecycle information integration using RDF and OWL to describe the templates of part7.

3. Modelling Geometric Objects

According to ISO15926, complex objects must be defined as templates, concepts that are defined using basic entities until they are reduced to basic terms (Proto and Core Templates). They must be compliant with ISO 15926 - Part2 and ISO 15926 - Part4, ensuring the integration of data portability and interoperability.

To define a best practice of how to represent geometry and topology of the manufactured and geological objects of an industrial process in ISO 15926, the ISO 15926 Part3 was created [ISO 15926-3 2007]. It presents a huge library of basic terms and definitions to be used for modeling.

According to ISO 15926, the geometric objects and properties must be modelled using Templates (ISO 15926-Part7). They are defined by decompositions of terms into simpler ones, in finite steps, until they are reduced to basic (or primitive) geomet-

ric terms. These basic terms (Core Templates) must be ISO 15926-Part2 compliant [Silva and Lopes 2011].

We present three approaches of modeling geometric objects, regarding a circle (geometric entity) as an example of how to use them.

3.1. Identification of ISO 15926-Part3 Elements

In the modeling process, it is important to understand the requirements of the object that will be modelled (stage 1). At this moment we will identify the object (e.g. circle) properties according with the ISO 15926-Part3 [ISO 15926-3 2007]. All the entities definitions present in this work were extracted from the ISO 15926-Part3. Any term in boldface represents a term in the ISO 15926 ontology. Circle definition:

> *An object is a **circle** if and only if: 1-it is **curve**; 2-it lies in a **plane**; 3- there is a centre point that is equi-distant from each point in the curve. NOTE 2 A **circle** has the geometric properties: radius; center and plane. These properties can be given for a **circle** by a **axial_reference_placement** and a **radius**. A **circle** has two alternative values for the **axial_reference_placement** corresponding to opposite directions for the normal.*

According with the definition, the concept **circle** is subclass of the concept **curve** and it is defined by a **radius**, a central point and a plane. So, the properties of a **circle** can be defined by the concepts **radius** and **axial_reference_placement**.

> *An object is a **radius** if and only if:1-it is a function between geometric objects with a unique **radius** and **metric_space_length**; 2- it specifies the radius. An object is an **axial_reference_placement** if and only if: 1-it is a function between geometric objects with a unique axial placement and **axis1_placement** (which is a **metric_space_point** and a direction denoted z); 2- it specifies the position and orientation of the geometric object.*

The concept **radius** is defined by a **metric_space_length**, that stores the measure of the radius. So the concept **radius** is used to link the measure with an object that has a **radius**, at this case the **circle**.

By the definition, the concept **axial_reference_placement** is used to connect a plane with an object. This plane is defined by the concept **axis1_placement**, that is composed by a set of points (**metric_space_point** and one direction (**direction**). Then the concept **axial_reference_placement** will connect the **axis1_placement** and the **circle**.

3.2. Identification of the Necessary Templates in Part7

After the requirements are known, it is necessary to analyze the data and the relationships that will be used in the modeling process (stage 2) [Kim et al. 2011]. In ISO 15926, the first step is to look for the concepts and relationships (templates) that will be used to model the object, ensuring that they are defined either at the Reference Data Library (RDL) or the Template Library (TPL) [Association 2008]. If it does not exist, it is necessary to ask to PCA [POSC Caesar Association 2012b] or its Special Interest Groups [POSC Caesar Association 2012a] to add it to the databases.

During the circle's modelling process, it was observed that some concepts were not connected with each other. By the Fig. 2, only the **circle** is connected with the class

curve, because **circle** is subclass of **curve**. In its definition, the concept radius is part of a circle, but it is not a **circle** (analogous to **axial_reference_placement**), so it is necessary to compound this relation. The compositions of these relations will be done with the construction of templates, whose methodology is described by the document ISO 15926 Part7 [ISO 15926-7 2011]. The templates hide the internal complexity of the models (described by the axioms), since access is given by the elements present in the signature.

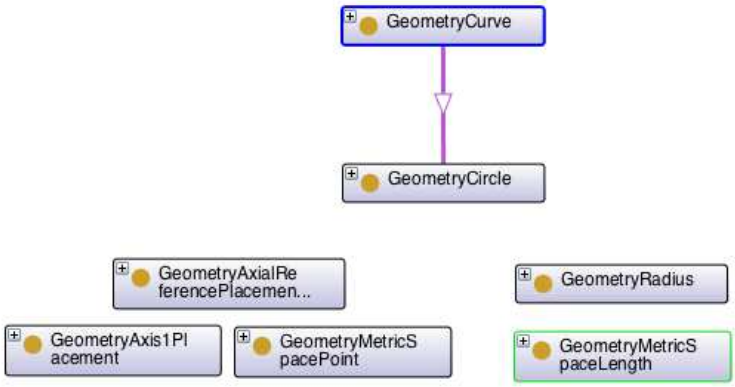


Figure 2. Identified Classes at Part3

The modeling process of a template has two steps:

1. Definition of the signature, that describes the elements that compound the relationship;
2. Definition of Axioms/Sentences in First Order Logic (FOL), that describes the semantics through the relations between the elements presented in the signature.

The axioms will be used to verify the consistency of the template. This verification is done by a tool called Template Expander that expands the axiom until the description with concepts defined in ISO 15926-Part2 or ISO 15926-Part4 [ISO 15926-7 2011].

The specification of a template axiom in FOL is done with the *if and only if* logical connective, where the signature of the template is on the left side, and the sequences of formula connected with the conjunction connective are on the right side.

Example: The template **RealMagnitudeOfProperty** is used to connect a concept classified as a **property** with a numeric value and a **scale** (as in Table 1 and the following axiom).

Table 1. Signature of RealMagnitudeOfProperty

Order	Rule	Type
1	hasProperty	Property
2	valPropertyValue	ExpressReal
3	hasScale	Scale

```
RealMagnitudeOfProperty(x1, x2, x3) <->
    property(x1) & ExpressReal(x2) & scale(x3) &
    exists u (MagnitudeOfProperty(x1, u, x3) & IdentificationByNumber(x2, u) ) .
```

What follows is a research about the template modeling process, regarding a circle as an example.The first approach presents a simplified modeling process. As some properties of the model are hidden due to its simple construction, it is necessary to understand the full model to infer these properties by queries. In the second approach, the model has more properties explicit and therefore the modeling process is more difficult it is possible to access the properties with simpler queries (it is not needed to know the full model to infer the properties in queries). The third approach proposes an intermediate abstraction between the first and the second one.

Alternative 1: Easy to Model but Difficult to Query. As defined by the ISO15926-Part3, **axial_reference_placement** and **radius** are functions that connect concepts, so they are candidates for templates.

In this alternative, the model of the template has a low granularity, it hides some possible templates without compromising the model structure. We constructed three templates: **RadiusTemplate**, **MetricSpacePointTemplate** and **DirectionTemplate**. The first template will connect one object that has a radius with a value that describes the length of radius. Its signature is shown at Table 5

Table 2. Signature of RadiusTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithRadius
2	hasRadius	RealNumber
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber

The parameters of the template signature above are of an object that has a radius (**circle**), the radius value and the lower and upper bounds of a scale.

```
RadiusTemplate(x1, x2, x3, x4) <->
    ObjectWithRadius(x1) & exists m radius(m) & hasEnd1(m,x1) & hasEnd2(m,k)
    exists k metric_space_length(k) & exists j scale(j) &
    exists l ( PropertyRange(l) & LowerUpperMagnitudeOfPropertyRange(l, j, x3, x4) &
        RealMagnitudeOfProperty(k, x2, j) ).
```

The concept **metric_space_length** alone does not represent the numeric value of a radius, it defines just a measure. A relationship between this measure and the circle is done by the template **RadiusTemplate** that relates this measure with a scale.

Some of the templates that are necessary to the modeling process can be found at the ISO 15926-Part7. In the template proposed above, the templates **RealMagnitudeOfProperty** (see Table 5) and the **LowerUpperMagnitudeOfPropertyRange** are at ISO 15926-Part 7.

According to ISO 15926-Part3, **metric_space_length** is subclass of **property**. Thus, it satisfies the condition of the template **RealMagnitudeOfProperty**. The template **RealMagnitudeOfProperty** claims an **scale** object, defined by the ISO 15926-Part2 . The scale is used to define a range of allowed values. To model a scale the template **LowerUpperMagnitudeOfPropertyRange** is necessary to connect two values with a scale, that is connected with a numeric value and a **metric_space_length**.

The template **MetricSpacePointTemplate** will connect an object with a **metric_space_point** with three real values (Table 3) that defines a plane according to ISO 15926-Part3.

Table 3. Signature of MetricSpacePointTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithAxialReferencePlacement
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

```
MetricSpacePointTemplate(x1, x2, x3, x4) <->
    ObjectWithMetricSpacePoint(x1) &
    exists c(CoordinateSystem(x1, c) & ListOfReals3Template(c, x2, x3, x4)).
```

The template CoordinateSystem, presented at ISO15926-Part7, specifies a plane with its three coordinates, that are connected with the template **ListOfReals3Template**.

The template **DirectionTemplate** connect three real values to represent the direction of the object (see Table 4).

Table 4. Signature of DirectionTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithDirection
2	hasDirectionX	RealNumber
3	hasDirectionY	RealNumber
4	hasDirectionZ	RealNumber

```
DirectionTemplate (x1, x2, x3, x4) <->
    ObjectWithDirection(x1) &
    exists c ( CoordinateSystem(x1, c) & ListOfReals3Template(c, x1, x2, x3) ).
```

Bellow is presented a graphic example of templates instantiations (**RadiusTemplate**, **MetricSpacePointTemplate**, **DirectionTemplate**) to construct the circle with a radius which the value is 3, with the position(1,2,3) and the direction expressed by the coordinate (1,0,0). It uses the following diagram language. The example is in Fig. 4. All the following Figures follows the legend in Fig. 3.

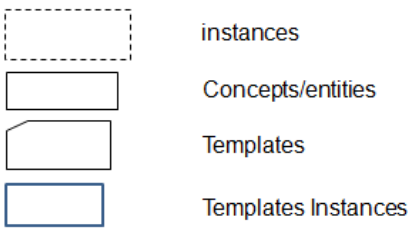


Figure 3. Legend of diagrams

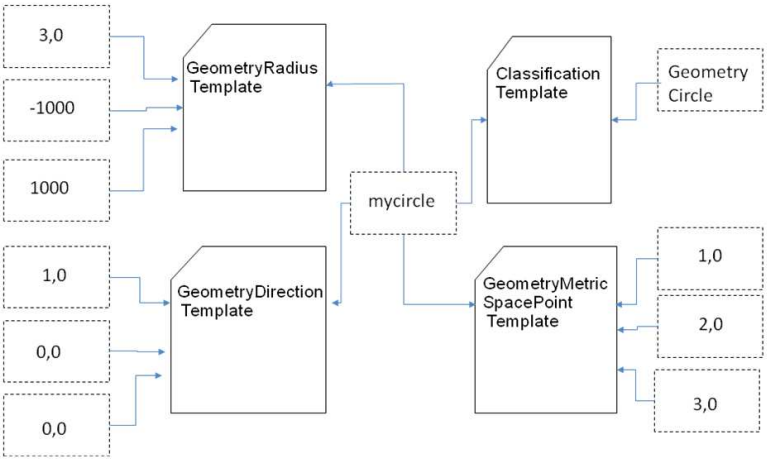


Figure 4. Instantiation example of alternative 1

Alternative 2: Hard to Model but Easy to Query. In this alternative, the granularity is high. The model is defined with five templates. As the first alternative, all the properties that define a **circle** are modelled. This process is more complex, but once it is modelled and instantiated, the queries about any properties will be done with ease.

The template **RadiusTemplate** (Table 5) is used to join all the properties about a **radius**. Its signature is the same as that of alternative 1, but the axiom that describes the template model is more detailed.

Table 5. Signature of RadiusTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithRadius
2	hasRadius	RealNumber
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber

```
RadiusTemplate(x, y, z, w) <->
    ObjectWithRadius(x) & RealNumber(y) & RealNumber(z) &
    RealNumber(w) & exists m ( radius(m) hasEnd1(m,x1) &hasEnd2(m,k) ) &
    exists k (metric_space_length(k) &
        exists  j (Scale(j) & exists l (PropertyRange(l) &
                LowerUpperMagnitudeOfPropertyRange(l, j, z, w) &
                RealMagnitudeOfProperty(k, y, j))) &
        exists p (MappingTriple(m, x, k) & radius(p)) ) .
```

In the formula above, the template **MappingTriple** [ISO 15926-7 2011] joins the object with **radius** to its properties.

The template **AxialReferencePlacementTemplate** (Table 6) defines the circle's plane. It relates six real values: three that define the **ReferencePoint** and three others that defines the **Direction**.

```
AxialReferencePlacementTemplate(q,px,py, pz, dx, dy, dz) <->
    ObjectWithAxialReferencePlacement(q) & exists k(axis1_placement(k) &
        ReferencePointTemplate(k, px, py, pz) &
        ReferenceDirectionTemplate(k, dx, dy, dz) &
        exists p(MappingTriple(p, q, k) & axial_reference_placement(p))).
```

Table 6. Signature of AxialReferencePlacementTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithAxialReferencePlacement
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber
5	hasDirectionX	RealNumber
6	hasDirectionY	RealNumber
7	hasDirectionZ	RealNumber

The template **AxialReferencePlacementTemplate** uses the **ReferencePointTemplate** (Table 7) and **ReferenceDirectionTemplate** (Table 8). These templates define the reference point and the direction respectively.

Table 7. Signature of ReferencePointTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithReferencePoint
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

Table 8. Signature of ReferenceDirectionTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithReferenceDirection
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

```
ReferencePointTemplate(x, px, py, pz) <->
    ObjectWithReferencePoint(x) & exists k ( metric_space_point(k) &
        exists c(CoordinateSystem(k,c) & ListOfReals3Template(c,px,py,pz)) &
        exists p (MappingTriple(p, x, k) & reference_point(p)) ).

ReferenceDirectionTemplate(x, dx, dy, dz) <->
    ObjectWithReferenceDirection(x) &
    DirectionScaleTemplate(x, dx, dy, dz) .
```

The templates **CoordinateSystem** and **ListOfReals3Template** have the same semantics of the templates presented at the alternative 1. The template **ReferenceDirectionTemplate** uses the template **DirectionScaleTemplate** (Table 9), that connect the three real values using the template **ListOfReals3Template** which has the same semantics presented at the alternative 1.

```
DirectionScaleTemplate(x, dx, dy, dz) <->
    ObjectWithDirection(x) & exists k ( direction(k) &
        exists c(CoordinateSystem(k,c) & ListOfReals3Template(c,dx,dy,dz)) &
        exists p ( MappingTriple(p, x, k) & direction_scale(p)) ).
```

Table 9. Signature of DirectionScaleTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithDirection
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

Bellow is presented a graphic example of templates instantiations (Fig. 5), that uses the same diagram language and values of the circle instantiation proposed at the alternative 1: a circle with a radius which the value is 3, with the position(1,2,3) and the direction expressed by the coordinate (1,0,0).

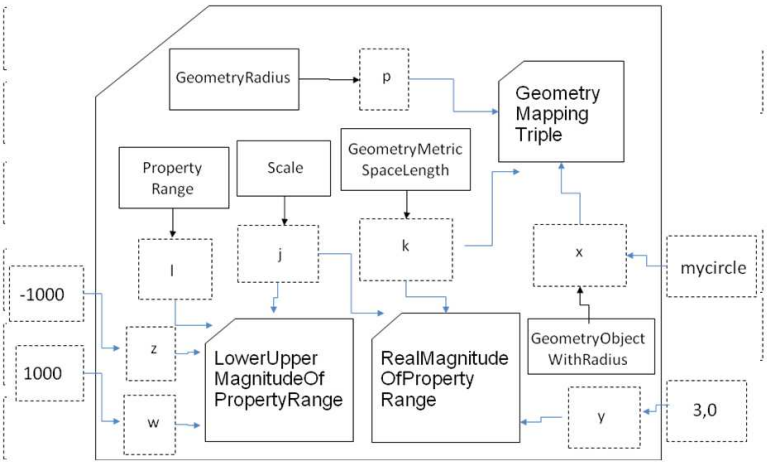


Figure 5. Instantiation example of alternative 2

Alternative 3: Trying a Balanced Approach. This approach presents two alternatives of intermediate granularity using a high level template. It aims to propose a model whose properties are explicit, to simplify the process of constructing queries in alternative 1 and simplifying the model construction in alternative 2, using all of its elements. In our first attempt, called **CircleTemplateAlternative3_1**, a higher level of abstraction is presented to construct instances of the model in the alternative 1. It uses all of its elements with a new template that constructs all the necessary instances, as a procedure that encapsulates all the process. This template signature is present in Table 10. The **CircleTemplateAlternative3_2** presents a more compact process of individuals construction which does not affect query construction negatively. Its template signature is shown in Table 10. The signature is the same for both alternatives in this approach, differing in its axioms, as follows.

```
CircleTemplateAlternative3_1(q,rd,rl,ru,px,py,pz,dx,dy,dz)<->
    circle(q) & RadiusTemplate(q, rd, rl, ru) &
    MetricSpacePointTemplate(q, px, py, pz) &
    DirectionTemplate(q, dx, dy, dz).
CircleTemplateAlternative3_2(q,rd,rl,ru,px,py,pz,dx,dy,dz)<->
    circle(q) & RadiusTemplate(q, rd, rl, ru) &
    AxialReferencePlacementTemplate(q,px,py,pz,dx,dy,dz).
```

Table 10. Signature of CircleTemplateAlternative3_1 and CircleTemplateAlternative3_2

Order	Rule	Type
1	hasPossessor	Circle
2	hasRadius	metric_space_length
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber
5	hasPositionX	RealNumber
6	hasPositionY	RealNumber
7	hasPositionZ	RealNumber
8	hasDirectionX	RealNumber
9	hasDirectionY	RealNumber
10	hasDirectionZ	RealNumber

4. Conclusions

The effort in the development and application of the ISO 15926 standard contributed with a new paradigm of information management for the Oil e Gas industry, that will reduce the costs in this area [Gallaher et al. 2004]. For the development of computer systems that are compliant with the standard across the industry, it shall know how to define, to manage, to extend the information models to store the data in a neutral format. There are many documents about the ISO15926 standard, but is difficult to organize the knowledge and to understand how to model the concepts without a methodology. It creates barriers for the deployment of the standard. Collaborating on this challenge, this work presents three alternatives that can be adopted at the modelling process. The two first alternatives have different levels of information granularity and one should be adopted depending of the queries to the Endpoints to retrieve the information. The last alternative uses high level templates to encapsulate the process of linking the elements at the instantiation of the others templates.

In future works, the main objective is to develop the standard researching subjects as the implementation of tools to help domain experts use the ISO 15926 standard, i.e. software to model and verify ISO 15926 templates, as well as an environment to create and to manage distributed data bases built upon the ISO 15926 proposed paradigm, building on the accumulated experience of the iRING User Group etc.; Implementation of the models using Web Ontology Language using the ISO 15926-Part8, envolving studies correlated with present day ontology challenges such as: how to store the ontology, how to manage the RDF triple store, how to make an efficient query across distributed RDF databases on the web; Design of an architecture to support format neutral exchange of 2D and 3D documents, based on SPARQL Endpoints providing federated management of process plant item symbology and Engineering document templates.

Acknowledgements

The authors thank TecGraf/PUC-Rio Computer Graphics Technology Laboratory and CNPq, for supporting this work, the PCA Geometry Special Interest Group and most especially Mr. Onno Paap for his advice and encouragement.

References

Association, P. C. (2008). RDS/WIP. http://rdl.rdlfacade.org. Acessed: February 2012.

Batres, R., West, M., Leal, D., Price, D., and Naka, Y. (2007). An upper ontology based on ISO 15926. *Computers & Chemical Engineering*, 31(5-6):519–534.

Consortium, T. W. W. W. (2004). Web ontology language overview. http://www.w3.org/2004/OWL/. Acessed: April 2012.

d'Aquin, M. and Noy, N. F. (2012). Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:96–111.

Gallaher, M. P., O'Connor, A. C., Jr., J. L. D., and Gilday, L. T. (2004). Cost analysis of inadequate interoperability in the US capital facilities industry. Technical report, National Institute of Standards and Technology.

IRING User Group (2012). IRING user group. http://iringug.org/wiki/index.php?title=Main_Page. Acessed: February 2012.

ISO 10303-1 (1994). Overview and fundamental principles.

ISO 10303-104 (2000). Integrated application resource: Finite element analysis.

ISO 10303-42 (2003). Integrated generic resource: Geometric and topological representation.

ISO 15926-1 (2004). Overview and fundamental principles.

ISO 15926-2 (2003). Data model.

ISO 15926-3 (2007). Reference data class.

ISO 15926-4 (2007). Initial reference data.

ISO 15926-7 (2011). Implementation methods for the integration of distributed systems: Template methodology.

Kim, B. C., Teijgeler, H., Mun, D., and Han, S. (2011). Integration of distributed plant lifecycle data using iso 15926 and web services. *Annals of Nuclear Energy*, 38(11):2309–2318.

Lee, S., Han, S., and Mun, D. (2012). Integrated management of facility, process, and output: data model perspective. *Science China-Information Sciences*, 55(5).

Pawsey, N. (2010). Iso 15926 & interoperability. In *PCA Meeting*.

POSC Caesar Association (2012a). PCA geometry special interest group. https://www.posccaesar.org/wiki/SigGeometry. Acessed: April 2012.

POSC Caesar Association (2012b). POSC caesar trac. https://www.posccaesar.org. Accessed: April 2012.

Silva, G. M. H. and Lopes, G. B. M. (2011). An approach about the modelling process to geometric objects with the ISO 15926 standard. In *Annals of CIB-W102 – Information and Knowledge Management*.

Applying Graph Partitioning Techniques to Modularize Large Ontologies

Ana Carolina Garcia, Letícia Tiveron, Claudia Justel, Maria Cláudia Cavalcanti

Seção de Engenharia de Computação. Instituto Militar de Engenharia

Praça General Tibúrcio, 80 Praia Vermelha – Urca, Rio de Janeiro, RJ – Brazil

{carolina.acgg, leticiativeronbt}@gmail.com, {cjustel, yoko}@ime.eb.br

Abstract. *Nowadays, it is difficult to reuse ontologies, especially those that cover a large domain. It is in this context that ontology modularization can be useful. The goal of this work is to investigate graph partitioning techniques and their application on the modularization of large ontologies, typically, biomedical ontologies. Such investigation may be divided in two steps: (i) how to convert an ontology, represented in OWL or RDF languages into a graph; (ii) which partitioning algorithm would be suitable. More specifically, this work focus is on how to preserve certain ontology properties/relationships in the generated modules. Therefore, a single way of graph conversion was adopted and user-defined edge weights were taken into account. Five graph partitioning algorithms were used for the present investigation, but just three of them were used to verify their behavior in face of edge weight variations. A case study was conducted using a toy-ontology on the pizza domain, and showed preliminary but interesting results.*

1. Introduction

The constant growth of data and publications in the biomedical area has been pushing the creation and reuse of domain ontologies in that area, not only for structured data annotation, but also for text indexation and annotation. Examples of such reusage are: Genbank[1], Pubmed[2] and NCBO Portal[3]. Genbank is the most popular comprehensive database that contains publicly available nucleotide sequences, for more than 380,000 organisms. Each sequence feature (genes, repetitive areas, etc.) is usually annotated with ontology references. Pubmed is one of the most popular digital biomedical citation reference (more than 21 million). Each text citation is associated (indexed) using the MeSH [4] thesaurus. A more detailed indexation, also known as text annotation, associates text expressions to ontology terms. The NCBO (*The National Center for Biomedical Ontology*) BioPortal provides the Annotator tool, specially created to support biomedical text annotations.

The Open Biological and Biomedical Ontologies (OBO) Foundry[5] *and the* NCBO BioPortal provide together more than 300 ontologies. How can a biomedical annotator deal with such a variety of ontologies? In addition, it is even more difficult

[1] http://www.ncbi.nlm.nih.gov/genbank/
[2] http://www.ncbi.nlm.nih.gov/pubmed/
[3] http://bioportal.bioontology.org/
[4] http://www.nlm.nih.gov/mesh/
[5] http://obofoundry.org/

to reuse them taking into account that typical biomedical ontologies have more than five hundred terms.

It is in this context that ontology modularization can be useful. However, in order to put modularization into practice greatly depend on the goals that are pursued, and thus, the splitting of ontologies into smaller modules has to follow some criteria [Parent and Spaccapietra, 2009]. It is worth noting that ontologies are semantic-based structures, in which each class and each property have different meanings. Such differences should be taken into account in the process of modularization, i.e., certain classes and/or properties (relationships) may be more relevant than others in the generated modules. For instance, for a user of the Gene Ontology (GO) [GO Consortium, 2000] it may be more important to generate modules where part-of relationships between selected nodes are all included. Therefore, a good criterion for generating reusable ontology modules is to rank properties, so that they are not discarded during modularization.

The task of modularizing large ontologies, typically, biomedical ontologies, can be divided in two steps: (i) how to convert an ontology, represented in OWL or RDF languages into a graph; and (ii) which partitioning algorithm would be suitable. With respect to (i), there are different ways of doing such conversion, as stated in [Coskun *et al.*, 2011], but either one should represent property ranking values in the graph. With respect to (ii), there are many graph partitioning algorithms, such as, *Spin Glass, Fast Greedy, Walktrap, Leading Eigenvector* and *Edge Betweenness*. However, just some of them take into account edge weights: *Spin Glass, Walktrap* and *Fast Greedy*. In this context, a question still remains: which of these algorithms would be more suitable for the ontology property-aware modularization problem?

There are previous works that evaluate partitioning algorithms applied to ontologies [Coskun *et al.*, 2011][Oh and Yeom, 2012], but they did not take into account edge weight variations.

This paper investigates the behavior of graph partitioning algorithms with respect to edge weight variations, and describes a case study that shows some initial but interesting results. In order to conduct this study it was necessary to adapt and use existing tools. For (i), the PATO[6] tool was used because it includes an ontology-graph conversion mechanism that adds weights to two types of properties (is-a and other domain properties). It had to be adapted in order to assign a different weight to each distinct domain property. Then, for (ii), the iGraph[7] tool was chosen as it includes implementations of three edge weight aware partitioning algorithms.

The case study was carried out using a toy-ontology on the pizza domain. Although the modularization targets are large ontologies, it would be very difficult to analyze and evaluate the resulting modules for these ontologies. Therefore, the idea of using a small ontology was to facilitate the analysis of the modularization results. Moreover, the choice for a common knowledge domain, such as pizza, was to avoid the need for biologists or domain specific specialists in the initial tests. The pizza case study showed that the variation of properties weights led (according to the user needs) to different partitioning results. It was noted that some algorithms kept most of the

[6] http://web.informatik.uni-mannheim.de/anne/Modularization/pato.html
[7] http://igraph.sourceforge.net/

highest weight properties within the modules and did not use them as cut edges (between the modules).

The rest of this paper is organized as follows. The following section describes the biomedical scenario, which motivates this investigation. The third section describes briefly some of the main graph partitioning techniques. The fourth section describes the experiment and discusses its results. The fifth section concludes the paper, pointing to future work.

2. Biomedical Ontologies

Nowadays there are many ontologies on the biomedical domain that can be found at *The Open Biological and Biomedical Ontologies (OBO) Foundry* and at the NCBO BioPortal. The OBO Foundry is maintained by a group of researchers that establish a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain [Smith et al. 2007]. Besides being a repository, the OBO plays the role of a reference organism, which reviews and certifies a set of ontologies on the biomedical domain. At the time of the writing of this paper, there were around 113 ontologies, from which only 8 were considered OBO ontologies, while the other 105 were still candidate ontologies. The GO (Gene Ontology) is one of the most popular among OBO ontologies.

The NCBO BioPortal [Noy et al. 2009] is an ontology repository that feeds a text annotation service. At the time of this writing there were 306 ontologies. After a quick analysis about their size, it was possible to conclude that approximately 3% have more then 100,000 classes; 8% have more than 10,000 classes; and more then 50% have more then 500 classes. Then, it is fair to say that in the biomedical domain, ontologies are typically of medium and large size. How can a biomedical (ontology-driven) annotator deal with such a variety of large ontologies?

This scenario motivates us on the investigation of alternative solutions for reducing the complexity of ontology reuse. The next section summarizes some of the graph partitioning techniques that could be useful to facilitate their reuse.

3. Graph Partitioning Techniques

The graph partitioning problem has been used to model problems of different areas. The growth and rapid evolution of real networks, created from technological and social networks, resulted in an increasing volume of data sets. These data can be used to extract information of the network elements. A network consists of a combination of elements and relationships between pairs of elements. Given this definition, we can construct an associated graph $G = (V,E)$, in which the vertices $(v \in V)$ represent the network elements and the edges $(e \in E)$ represent some kind of relationship between the elements.

The techniques for solving the graph partitioning problem try to divide the set V (the vertices of G) in clusters (also communities or partitions) that optimize a certain criterion. For instance, each cluster must have edges between internal vertices with high weight and edges between different clusters with low weight. Following this optimization criterion, the graph partitioning problem is NP hard and can be formally defined as:

Given a graph $G=(V,E)$, find p subsets $V_1, V_2, \ldots V_p$ such that:

i. $\bigcup_{i=1}^{P} V_i = V \ and \ V_i \cap V_j = \emptyset,$ for any $i \neq j$.

ii. $W(i)$ and W represent the sums of the weights of the edges between vertices inside the sets V_i and V, respectively.

iii. The sum of the weights of the edges that connect the vertices into two subsets V_i and V_j, for all pair i,j must be minimal.

There are several approximate algorithms to solve the graph partitioning. In this paper we consider the partitioning algorithms implemented in the library iGraph. This library provides an implementation of 5 (five) different algorithms for graph partitioning: *Edge Betweenness Community*, *Walktrap Community*, *Fast Greedy Community*, *Spin Glass Community* and *Leading Eigenvector Community algorithms*.

The *Edge Betweenness Community* is a divisive algorithm. It removes recursively edges of the graph until determine communities [Coskun et al., 2011]. From a non-weighted graph $G=(V, E)$ the *betweenness* of an edge $e \in E$, is the number of shortest paths that connect any two vertices v_1 and v_2 of V passing through the edge e. Note that there may be more than one shortest path between two vertices. In this case, if there are k shortest paths between the vertices v_1 and $v_2 \in E$, then each one will have a weight $1/k$ to calculate the edge's *betweenness* of these paths [Schaeffer, 2007].

This algorithm computes the values of the *betweenness* of each edge. And is based on the following observation: edges with higher value of *betweenness* must be connecting vertices of two different partitions, i.e., they are not inner edges in a partition. Then the algorithm divides the graph into clusters, removing one by one the edges with the highest value of *betweenness*. If more than one edge has the highest value, one is chosen randomly. After each removal, the *betweenness* is recalculated for each edge. This process is repeated until a stop criterion.

The *Walktrap Community* is an algorithm based on the following statement: "random walks in a graph tend to get trapped in dense parts of the graph, corresponding to the communities" [Coskun et al., 2011]. That is, by drawing a random path between two nodes, the nodes that belong to the path are more likely to belong to the same community. It is a hierarchical agglomerative algorithm, because the communities are built step by step through the union of vertices to form communities. Initially the algorithm treats all vertices of the graph as communities of a single node. Then, at each step, two communities are joined, until the stopping criterion.

The *Fast Greedy Community* is an algorithm widely used to determine communities for non-directed and sparse graphs $G=(V,E)$ [Eom et al., 2009]. This algorithm is based on the concept of modularity of a partition $C=\{C_1, \ldots C_p\}$, $Q(C)=\Sigma_{1 \leq i \leq p} a_{ii} - a_i^2$, where a_{ii} represents the edges of the graph inside the set C_i and a_i the number of edges with one endpoint in the set C_i The *Fast GreedyCommunity* algorithm maximizes the value of the modularity in a greedy fashion.

Initially, the algorithm considers each vertex of the graph as a unitary community. Then, the algorithm finds the pair of communities C_p and C_q having the maximum value $\Delta Q_{ij} = e_{ij} - e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j)$, where e_{ij} is the number of edges between C_i and C_j. Next, the algorithm combines these two communities C_p and C_q in only one community. The process is repeated while ΔQ is a positive number. During the process of union of two communities into one, the algorithm updates the values

corresponding to neighboring communities (internal and external edges of the communities affected by the union of C_p and C_q).

The *Spin Glass Community* is an algorithm based in a thermodynamics technique to model the graph-partitioning problem. The meta-heuristic *Simulated Annealing* is used to solve the minimization energy problem considering in this model. In this context, the spin states consider the vertices of the graph and the structure of the partition in the graph is interpreted as the spin configuration that minimizes the energy of the Spin Glass.

The *Leading Eigenvector Community* algorithm uses the concept of modularity in a different way to perform the partitioning. In this case, the algorithm finds the eigenvector corresponding to the most positive eigenvalue of a modularity matrix, defined from values ΔQ_{ij} and divide the network into two groups, according to the signs of this vector elements [Newman, 2006].

Table 1 surveys 3 characteristics of the five algorithms mentioned before: type, worst-case complexity, possibility of use of weighted edges. Note that just three of them take into account edge weights. The complexities of these algorithms were obtained from the iGraph API documentation. We note the graph *G=(V,E),* where $|V|$ is the number of vertices and $|E|$ is the number of edges.

Table 1 - Comparison between the five algorithms.

Algorithm	Weighted Edges	Type	Complexity
Edge Betweenness	No	Divisive	$O(\|V\|^3))$
Walktrap	Yes	Agglomerative	$O(\|V\|^2 log(\|V\|))$
Fast Greedy	Yes	Greedy	$O(\|E\| + \|V\| * log^2(\|V\|))$
Spin Glass	Yes	Aproximate	Not Found
Leading Eigenvector	No	Spectral	$O(\|E\| + \|V\|^3)$

4. Case Study

As stated in [Parent and Spaccapietra, 2009], each module is expected to show a similar unit of purpose, gluing together those elements that participate on a given goal. A module should make sense to ontology engineers seeking to (re)use them [Grau et al., 2006]. For instance, a module should represent an agreed conceptualization of a sub-domain of the domain of the ontology.

Evaluating ontology modules is not an easy task. A recent related work [Oh and Yeom, 2012] proposes a new evaluation framework for selecting an appropriate ontology modularization tool. In their work, modularization tools were evaluated as use cases according to the proposed framework, which takes into account three aspects: tool performance, data performance, and usability. Data performance includes verifying the cohesion of a module, which means asking if a module contains plenty of concepts, relationships, and axioms. However, although different partitioning methods were compared, the property ranking was not used as a criterion for modularization evaluation.

Another related work [Stuckenschmidt and Klein, 2004] identifies ontology partitions depending on property weights. These weights are calculated based on graph dependencies (e.g. subclass, domain and range restrictions). However, a method purely based on the structure of the ontology may not be able to capture semantics of such properties.

The present work adopts a user-based, but probably not scalable approach. According to the user preference (weights assigned to object properties), a set of modules is generated. A module makes sense if it is cohesive, meaning if it includes the user-preferred properties, and the related concepts. Based on this criterion it was possible to evaluate the generated modules.

In order to verify the behavior of the selected five algorithms taking into account weighted ontology properties, a case study scenario was prepared: support tools were configured and/or adapted to execute such algorithms, having as input a chosen ontology. The following subsections detail the case study scenario and the subsequent executions of the algorithms, discussing the cohesion of the resulting modules.

4.1 Scenario Preparation

The modularization of a given ontology was divided into two tasks: the first one consists in representing a given ontology as a graph, and the second consists in partitioning this graph.

With respect to the first task, although it is a non-trivial task, it was not in the scope of this work to focus on this problem. There are different and richer ways of converting an OWL/RDF ontology into a graph representation [Coskun et al., 2011], but in the context of this work it was performed as follows: given an OWL file, converts it into a graph *G=(V,E),* where each OWL class or RDF resource corresponds to a vertex of *V*, and each OWL/RDF object property corresponds to an edge of *E*. In this type of conversion, usually there is no distinction between the edges, and therefore the different relationship types are lost. As we stated before, since ontologies are semantic-based structures and have different domain properties (object properties), the edge-weight variation is meaningful to their modularization.

The literature pointed to some tools to perform the first task. The Jena[8] java library and the PATO tool were alternatives. Their outputs are graphs in graphML and Pajek formats, respectively. Although Jena allows three distinct representations for the graph, PATO was more suitable as it allows assigning weight values to graph edges.

The PATO tool could be useful for the second task as well, since it performs the complete modularization of an ontology. However, this tool is poorly documented and it was not possible to identify the algorithm behind its partitioning algorithm. On the other hand, there were two known partitioning C++ libraries available: SNAP[9] and iGraph. SNAP provides the implementation of two different partition algorithms, *edge betweenness* and *fast greedy,* while iGraph provides three others besides those two: *spin glass, walktrap* and *Leading Eigenvector.* Furthermore, iGraph admits Pajek input format, allowing its use in conjunction with the PATO tool. Therefore, iGraph was chosen for its coverage and compatibility.

The PATO tool had to be adapted in two ways. First, it was removed all its unnecessary functionalities for our goal. Second, since PATO's original code only differentiates the subclass relationship and the domain properties, assigning the same weight to all the domain properties, it was necessary to adapt the code in order to allow assigning distinct weights to different domain properties.

[8] http://jena.sourceforge.net
[9] http://snap.stanford.edu/snap/

In addition to iGraph and PATO, the graphic interface of the R-tool[10] from iGraph was used in order to visualize the results. Also, it was developed a Java procedure whose input is the output of the iGraph library, in text format (.txt). For a partitioning that generates *k* communities, this routine returns k files (communities) in Pajek format.

To facilitate the analysis of the edge weight variation it was necessary to choose a domain, small but sufficient, ontology example. As most of the biomedical ontologies are large, they were initially discarded. Furthermore, it would be difficult to extract a reduced size module of it to work with. Therefore, work with a toy-ontology on a common knowledge domain would be a wise choice. A well-known example of toy ontology on the pizza domain (used in knowledge representation tutorials and courses) was chosen. However due to limitations of the PATO tool on dealing with OWL format, an RDF smaller version of the pizza ontology[11] was adapted and used.

4.2 Partitioning Results

The output graph obtained with PATO from the Pizza ontology is shown in Figure 1. This figure was generated with the aid of R-tool and Power Point. The Pizza ontology used has three types of properties: "*isa*", "*consists_of*" and "*consumes*" and for this study it was planned 5 (five) combinations of weight distribution for the edges, as described in the Table 2. From each choice of the edge weights combinations, 5 weighted graphs corresponding to the Pizza ontology were created.

Table 2 - Weight combinations of the edges assigned to the Pizza graph

Graph	consists_of	Consumes	Isa
0	1	1	1
1	1	1	2
2	1	2	1
3	2	1	1
4	2	2	1

	Vertex
0	Non_Vegetarian_Ingredient
1	Raw_Ingradient
2	Vegetarian_Ingredient
3	Vegan_Customer
4	Vegetarian_Customer
5	Pizza
6	Edible_Thing
7	Customer
8	Vegan_Ingredient
9	Small_Pizza
10	Medium_Pizza
11	Large_Pizza
12	Non_Vegetarian_Pizza
13	Vegetarian_Pizza
14	Vegan_Pizza
15	Non_Vegetarian_Customer

Graph

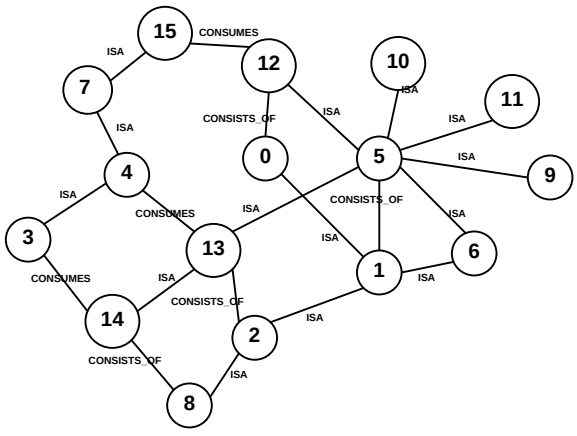


Figure 1 - Vertices of the graph corresponding to the Pizza ontology

[10] http://www.R-project.org/
[11] http://www.heiko-stoermer.net/teaching/2006-models-and-techniques-of-knowledge-representation

Each of the five resulting graphs was partitioned by *Fast Greed, Spin Glass* and *Walktrap* algorithms, which allows weighted graphs as input. The other two algorithms, *Leading Eigenvector* and *Edge Betweeness*, were executed only to the graph with constant weights (graph 0).

For graph 0, the *Spin Glass* and *Fast Greedy* algorithms obtained the same result (the same 3 communities). The *Walktrap* and *Edge Betweenness* algorithms obtained the same result (the same 3 communities). But the difference between the partitions obtained in these two cases is the allocation of vertices 0 and 12 (Non-Vegetarian Pizza and Non-Vegetarian Ingredient). Figure 2 shows the output by the 5 algorithms for graph 0. In this case, constant weights for the edges, the algorithms seem to have similar or slightly different behavior. However, the communities obtained with *Fast Greedy* and *Spin Glass* algorithms seem to be better.

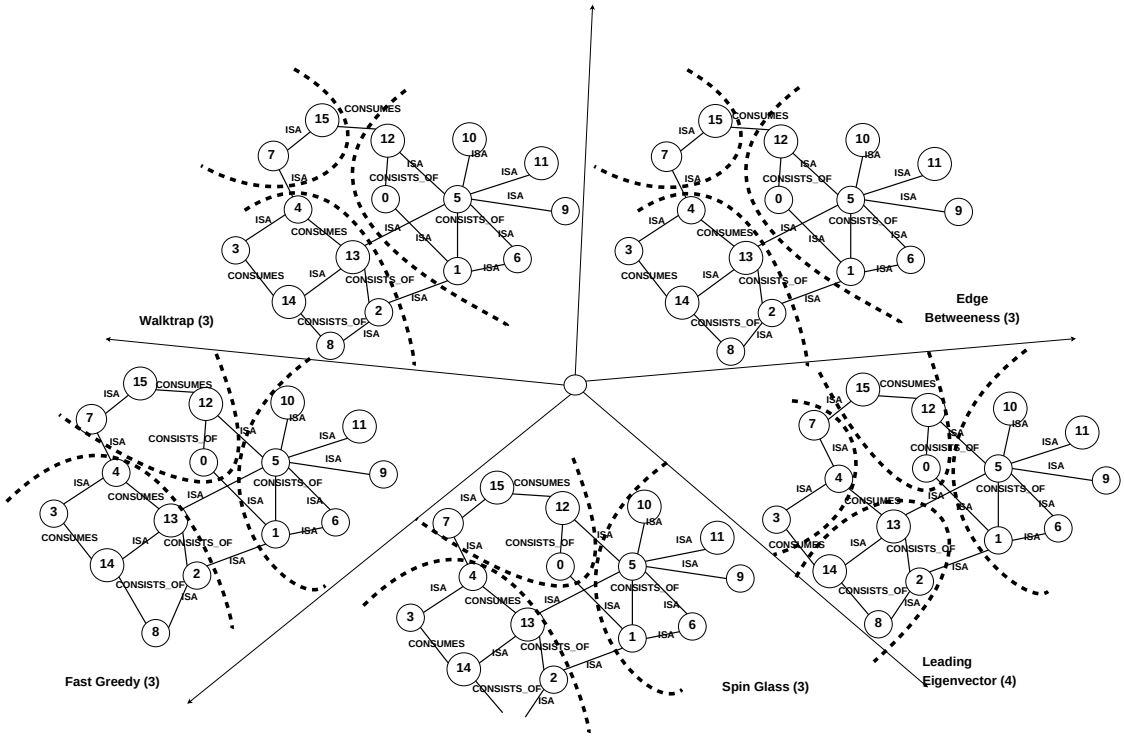


Figure 2 - Partitioning for the first graph (graph 0).

In what follows, we will discuss the content of each community generated with the algorithms. Table 3 shows the vertices allocated in each of the 3 communities by *Fast Greedy* and *Spin Glass* algorithms for graph 0. Both algorithms included vegan/vegetarians on one community and non-vegetarians in another. Furthermore, a third community of generic nodes, which did not fit in either first two communities, was generated. Note that vertex 7 is not allocated in the generic nodes community. However, vertex 7 is not adjacent to any other vertex in the third community, and therefore the obtained result makes sense.

For the graph 0, *Edge betweenness* and *Walktra*p algorithm obtain 3 communities. In this case, vertices 0 and 12 (Non-Vegetarian Ingredient and Non-Vegetarian Pizza) belong to the partition correspondent to general classes (Non-Vegetarian Customer). Both algorithms focus on paths (one is deterministic and the other is probabilistic), and this approach is different from the first two algorithms analyzed. However, more tests should be performed in order to obtain general conclusions.

Table 3 - Communities generated with *Fast Greedy* and *Spin Glass* algorithms for graph 0.

Community	Vertices	Description
Vegan/Vegetarian	2, 3, 4, 8, 13, 14	Vertices that represent Vegans and Vegetarians Pizzas, Ingredients and Customers
Non-Vegetarian	0, 7, 15, 12	Vertices that represent Non-Vegetarian Ingredient, Non-vegetarian Pizza and Non-Vegetarian Customer, and yet Customer (vertex 7)
Pizza	5, 11, 9, 10, 1, 6	Vertices that represent General classes (Pizza, Raw Ingredient, Edible Thing) and pizza sizes.

It is also worth noting that, different from the other algorithms, the *Leading Eigenvector* algorithm generated an extra partition with different types of customer (Vegan and Vegetarian Customer and Customer itself). Vertex 15 (Non-Vegetarian Customer), which seemed to be semantically closer to this community, was allocated in another related community, which includes Non-Vegetarian Pizza and Non-Vegetarian Ingredient.

With respect to the other graphs (graphs 1-4), *Fast Greedy* algorithm showed the best performance in terms of vertex clustering (communities). *Fast Greedy'*s partitioning results are shown in Figure 3, while Figures 4 and 5 show the partitioning results for the *Spin Glass* and *Walktrap* algorithms, respectively.

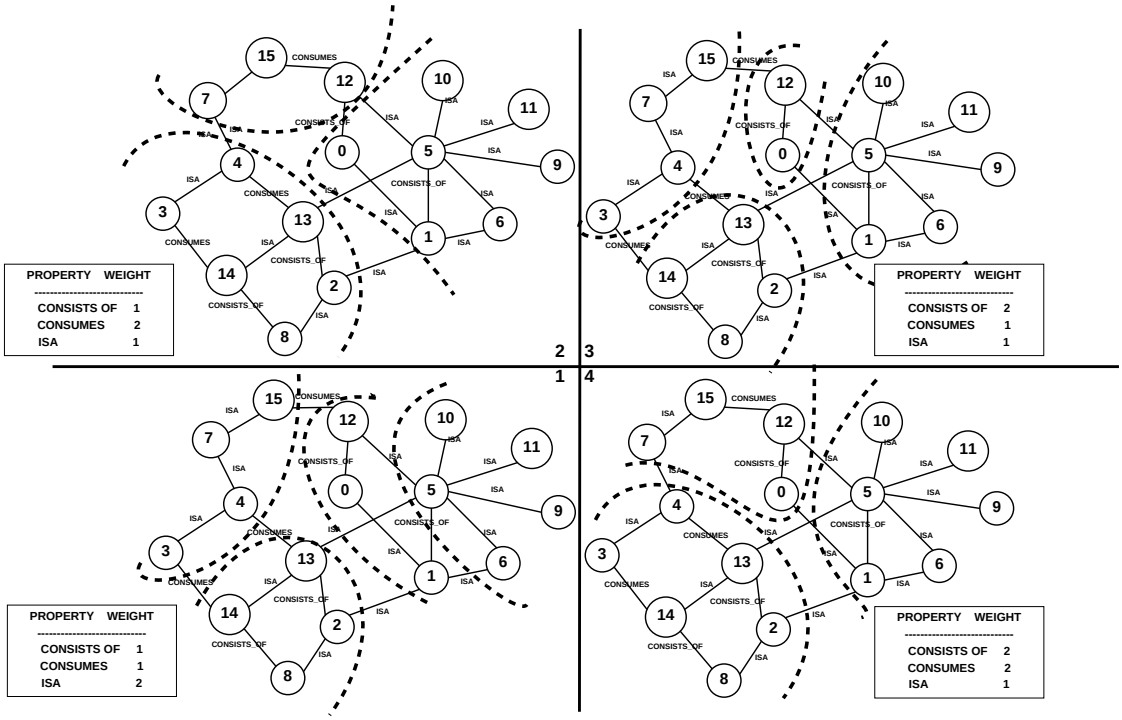


Figure 3 – *Fast Greedy* communities for different weight configurations (the numbers at the center correspond to graph numbers of Table 2).

Graph 3, which assigns the highest weight for the "*consists_of*" property, is the one for which *Fast Greedy* showed the best partitioning. This partitioning seems to make more sense than the one generated by graph 0 (analyzed previously). The four generated communities may be easily described, as follows: (i) Consumers (3,4,7,15); (ii) Non-vegetarians (12,0); (iii) Generic classes (1,5,6,9,10,11); (iv) Vegan and vegetarians (2,8,13,l4). Note that vertex 7 (Customer) now belongs to a community that includes the other customers. The partitioning for graph 4 is equal to graph 0. In

the other graphs (1 and 2) there are some out of place vertices. For instance, vertex 0 (non-vegetarian ingredient) is separated from vertex 12 (non-vegetarian pizza).

The partitioning results for Spin Glass algorithm do not show much variation. Graphs 2-4 partitioning results are equal to graph 0 results. Graph 1, which assigns the highest weight for the "isa" property, is the only one whose results are different, but interesting. Note that they are similar to graph 3 results of the Fast Greedy algorithm, with the difference that communities (ii) and (iii) are merged.

Similarly, partitioning results for Walktrap algorithm show variation only for graph 1, and one of its communities (0,1 and 6) do not make much sense, joining together vertices with not much in common.

Another important analysis is if the generated modules attend the property ranking criterion, i.e., if the module includes the properties and concepts according to the user priority assignment. As stated before, the idea is to prioritize one type of property, in order to maintain them in the resulting partition. Figure 3 shows that the partitioning results for graph 1, which assigns highest weight value to the "*isa*" property, shows that most of the edges that represent this property are "inside" the community, i.e., they are not between two different communities (cut edges). Similarly, partitioning results for graph 2, which assigns highest weight value to the "*consumes*" property, shows that all the edges that represent the "*consumes*" property are inside the communities. In other words, the vertices joined by edges of greater weight tended to remain in the same community, and these edges were then maintained in some partition. *Fast Greedy* algorithm also showed the best results, varying according to the different configurations of edge values. Table 4 summarizes its results. Note that when the "*consumes"* property is prioritized, it does not appear as a cut edge (0) in the graph. The same occurs with the *"consists_of"* property.

Table 4 – Property Priority (highest weight) versus the number of cut edge and inside properties for the *Fast Greedy* algorithm (analysis from Figure 3 graphs)

Priority (> weight)	cut edge properties			inside properties		
	isa	consists_of	consumes	isa	consists_of	consumes
Isa (graph 1)	3	2	3	11	2	0
Consists of (graph 3)	4	0	3	10	4	0
Consumes (graph 2)	4	1	0	10	3	3

5. Conclusion

This paper described an experiment that showed some initial but interesting results on how partitioning algorithms behave for ontology modularization, with focus on edge weight variations. In the context of ontologies, it makes sense to identify priorities for ontology object properties before partitioning. This can lead to more useful ontology modules from the user point of view.

The focus of this work was on the data performance evaluation, one of the aspects of the evaluation framework proposed in [Oh and Yeom, 2012]. More specifically, the focus was on the cohesion of the resulting modules. Five graph partitioning algorithms were executed for the graph representation of a toy-ontology on Pizza domain, but only three of them allowed the generation of weighted graphs. Among these three, *Fast Greedy* algorithm had the best preliminary results, showing "sensibility" with respect to the domain property ranking. However, further tests should be executed with larger and different ontologies. The suggested assumptions stated in this work can be refuted or confirmed by future work.

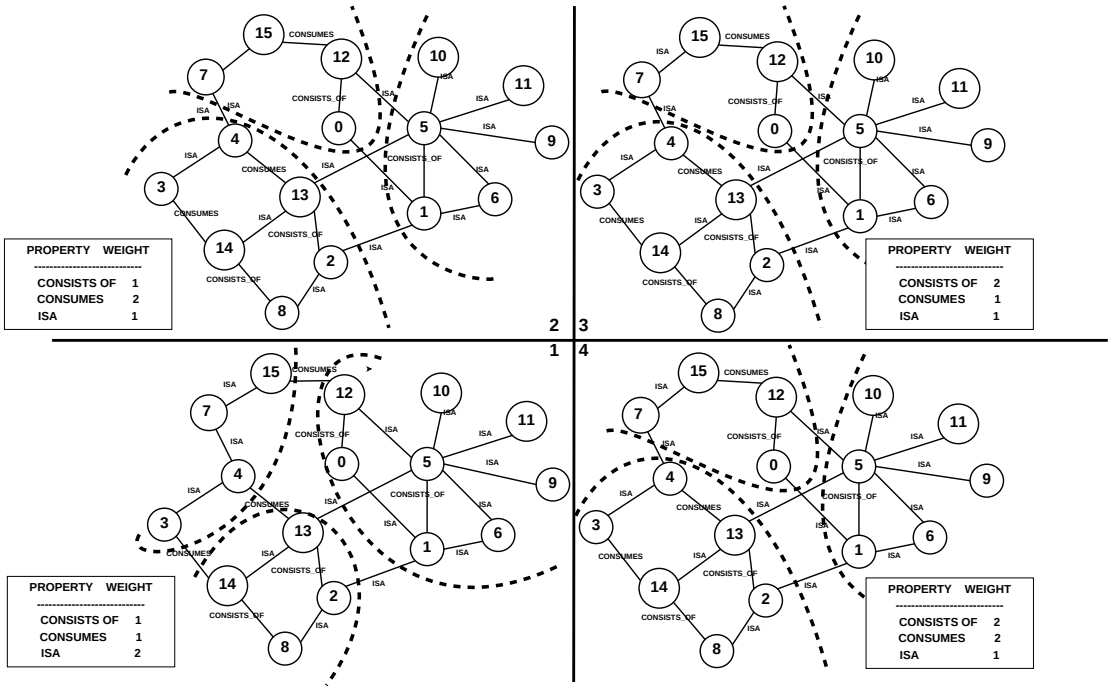


Figure 4 – *Spin Glass* communities for different weight configurations.

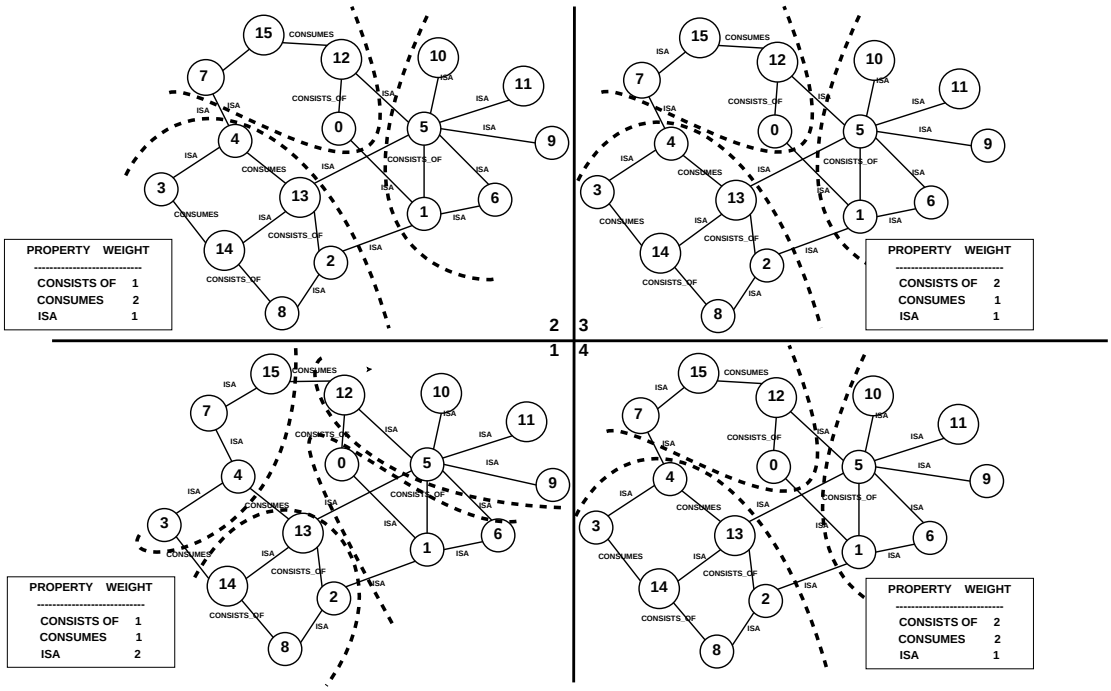


Figure 5 – *Walktrap* communities for different weight configurations.

Acknowledgements

The authors would like to thank CNPq (309307/2009-0; 305516/2010-8 486157/2011-3; PIBIC scholarships) and FAPERJ (E-26/111.147/2011) for partially funding their research projects.

References

Coskun, G.; Rothe, M.; Teymourian, K.; Paschke, A. (2011). "Applying Community Detection Algorithms on Ontologies for Identifying Concept Groups". Proc. of the Fifth International Workshop on Modular Ontologies, (WoMO 2011), p. 12-24.

GO Consortium. (2000) "Gene ontology: Tool for the Unification of Biology". Nat. Genet., 25(1), p. 25-29.

Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A. (2006) "Modularity and web ontologies". In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), p. 198-209.

Eom, Young-Ho; Choi, Yoonchan; Jeong, Hawoong; Kwak, Haewoon; Moon, Sue (2009) "Mining Communities in Networks: A Solution for Consistency and Its Evaluation". In: Proc. of the Internet Measurement Conf. (IMC 2009), p. 301-314.

Newman, M. E. J. (2006) "Finding community structure in networks using the eigenvectors of matrices". Physical Review E 74(3).

Noy, N. F., Shah, N.H., Whetzel, P.L. et al. (2009) "BioPortal: Ontologies and Integrated Data Resources at the Click of a Mouse". Nucleic Acids Research 37 (Web-Server-Issue), p. 170-173.

Parent, C. and Spaccapietra, S. (2009) "An Overview of Modularity". In: Stuckenschmidt, H., Parent, C.; Spaccapietra, S. (Eds) Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. Lecture Notes on Computer Science 5445, Springer, p. 5-23.

Schaeffer, S.E. (2007) "Graph Clustering". Computer Science Review 1(1), p. 27-64.

Smith, B., Ashburner, M., Rosse, C., et al. (2007) "The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration". Nature Biotechnology 25, p.1251-1255.

Stuckenschmidt, J. and Klein, M. (2004). Structure-based partitioning of large concept hierarchies. In: Proc. Int. Semantic Web Conference (ISWC).

Oh, S. and Yeom, H.Y. (2012) "A comprehensive framework for the evaluation of ontology modularization". Journal Expert Systems with Applications 39(10), p. 8547-8556

The Limitations of Description Logic for Mathematical Ontologies: An Example on Neural Networks

Fred Freitas, Fernando Lins

Informatics Center - Federal University of Pernambuco (CIn - UFPE)
Av. Prof. Luis Freire, s/n, Cidade Universitária, 50740-540, Recife – PE – Brazil

{fred,fval}@cin.ufpe.br

Abstract. *In this work, we discuss appropriate formalisms to account for the representation of mathematical ontologies, and, particularly, the problems for employing Description Logics (DL) for the task.DL has proven to be not expressive enough to such tasks, because it cannot represent, for instance, simple numerical constraints, except of cardinality constraints over instances of individuals and relation instances. Therefore, for such representations, we advocate the use of a more expressive formalism, based at least on first-order logic using a top ontology as support vocabulary. We also provide a thorough example on the representation of an Artificial Neural Network (ANN) defined in KIF (Knowledge Interchange Format), along with a discussion on the requirements needed for such a representation and the results achieved.*

1 Introduction

All In this work, we introduce a discussion on the appropriate formalism to carry out the task of representing mathematical knowledge, and particularly the problems for doing it with Description Logics (BAADER ET AL, 2003). It departed from an attempt to develop an Ontology of Multilayer Perceptron (MLP) Artificial Neural Networks (ANN). Our initial intent was to define it using the Semantic Web standardized description logic language OWL (Ontology Web Language) (WELTY ET AL, 2004), which would offer us a lot of benefits, like mature development tools, availability of reasoners, large community of users, etc.

On creating such ontology, our work was focused in developing a conceptual model that would include the necessary mathematics to enable the future development of knowledge-based applications that can reason over artificial neural networks, such as an intelligent agent capable of interacting with users to answer questions about neural networks that can even create and run them.

During the development of the ontology, we faced several representational problems while trying to represent the mathematical expressions inherent to neural networks in Description Logic (DL). Since this formalism is based on set theoretic semantics, it fits properly to representing domains in which relationships among sets suffice. For such domains, DL terminologies can accurately describe the domains' classes. On the one hand, the choice of this formalism for the Semantic Web lies actually on these grounds. On the other hand, it is not endowed with any apparatus to deal with many types of mathematical operations and constraints that would turn it into a more generic formalism, like what can be represented by Prolog, for instance. Moreover, a core mathematical ontology is required as a vocabulary to make for the needed complex well-defined mathematical meanings.

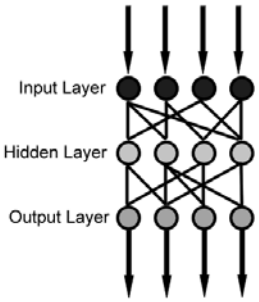
Therefore, for such representations, we advocate the use of a more expressive formalism, based at least on full first-order logic, and using a rich, well-defined support vocabulary. In order to illustrate the discussion, we first introduce the required basic mathematical definitions about MLP neural networks and the expressiveness requirements that a language should fulfill to represent the ontology. Then, we discuss the practical expressiveness limitations that hamper the use of Description Logic for such representations. Finally, we present our MLP ANN ontology and some of its more important definitions. We developed it in the Standard Upper Ontology Knowledge Interchange Format (SUO-KIF) language (PEASE 2009) and used the top ontology Suggested Upper Merged Ontology (SUMO) (NILES & PEASE, 2001) as support vocabulary, which proved well-tailored for our mathematical representations. We complete the article giving a very brief introduction of the SUO-KIF language and the top ontology SUMO, discussing some related work, future work and conclusions.Full papers must respect the page limits defined by the conference. Conferences that publish just abstracts ask for **one**-page texts.

2 An Example: Representing the MLP ANN in an Ontology

Artificial Neural Networks (ANNs) were developed as a rough abstraction of biological neural networks. An ANN consists of a number of nodes that perform simple numerical processing. Its nodes are highly interlinked and grouped in layers. Mathematically speaking, it implements complex function approximators (HAYKIN, 1994). ANNs are used in the computer mainstream to perform pattern learning and recognition.

The most commonly used ANN is the Multilayer Perceptron (MLP) (HAYKIN, 1994). It has three layers (entry, hidden and output) or more (when more than one hidden layer is used). Each layer consists of a set of "artificial neurons" connected with the neighbor layers, as displayed in the figure 1 below.

Figure 1– A MLP Artificial Neural Network, its layers and neurons.



A MLP ANN "learns" the implicit function it should approximate by computing, correcting and propagating classification errors throughout all its neurons, as follows. In the training phase, the MLP ANN is given a set of problem instances, usually represented as numerical values, along with their solutions, and process the data many times to assign correctly the neurons' parameters. Synapses among artificial neurons are modeled via activation functions, which trigger the delivery of numerical stimuli as inputs to the next layer's neurons. At the end of the learning phase, the network is tuned with the parameters so as to reflect as closely as possible the classification function implicit in the training set.

The Backpropagation algorithm (HAYKIN, 1994) is used to train the MLP ANN. It is divided into two phases. In the *forward phase*, the network is run and a classification output is calculated. In the backward phase, it compares the output with the correct result and, if wrong, it calculates and propagates the error through the weights of the connections among all the neurons of the network. Therefore, the learning ability of MLP ANNs lies on the neurons. The algorithm iterates between these two phases until it reaches a stopping criteria. These two phases are explained in detail in the next subsections.

2.1 Forward Phase

Here, the ANN basically computes summations of a product. At first, each neuron receives its entry and propagates it to the first hidden layer's neurons, if its activation function "triggers". For each neuron of the hidden layer, the potential of activation of the neuron j ($net_j^p, in\ the\ formula$) is calculated by summating all synapses weights (w_{ji}) from neurons i that reach neuron j, multiplied by the outputs of neurons ii:

$$net_j^p = \sum_{i=1}^{n} x_i^p w_{ji} \tag{1}$$

Then, the activation potential is applied to an activation function that will return the output of the neuron. One of the most used activation functions is the logistic function (also called sigmoid), that can be seen below:

$$P\left(net_j^p\right) = \frac{1}{1 + e^{-net_j^p}} \tag{2}$$

The output of the neuron, given by the result of the activation function, typically consists of a real value between -1 and 1. A synapse is defined as the propagation of the output as an entry to the next layer, or, if it is the last layer, one of the outputs belonging to the output vector of the network. Synapses in a neuron occur if its output is higher than a given threshold, usually set to 0. Synapses are then propagated (or not) through the network until they reach the output layer, finishing the forward phase. This phase is executed when the network is in the execution mode too.

2.2 Backward Phase

The first step of this phase consists in calculating the error between the output of the neuron j from the forward phase and the expected output from the instance of the training set, as in the following formula

$$\delta_j = \left(d_j - y_j\right) f'(net_j) \tag{3}$$

where d_j is the desired output, y_j the actual output, and $f'(net_j)$ the activation function of the neuron's derivate. Hidden nodes' errors are computed as follows:

$$\delta_j = f'(net_j) \sum_{i=1}^{n} \delta_l w_{lj} \tag{4}$$

They are the product of the activation function derivate of the current neuron and the summation of the products between the error (δ_j) and each of the last layers' weights (w_{lj}). Finally, the synapse weights that connect current neuron j with its antecessors (neurons i) are given by:

$$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j(t)x_i(t) \quad (5)$$

where $w_{ji}(t)$ stands for the old weight, $w_{ji}(t+1)$ means the new weight, η is the learning rate that regulates learning speed, $\delta_j(t)$ is the error and $x_i(t)$ is the input signal of the neuron with the old weight. The rough idea is slowly (as tuned by η) distributing the error issued by the network through all the layers and neurons.

In the next section, we enlist the requirements for representing such knowledge.

3 Requirements of Candidate Formalisms and Languages

To address the problem of choosing the knowledge representation formalism and language to represent the MLP ANN, we had to consider a number of representation requirements. We needed enough expressive power to represent the mathematical concepts on the ontology. Potential candidate formalisms and languages should, at the same time, allow us to do it in an easy way, as well as give us the possibility of computing results afterwards. Below, we list the most important requirements for that:

- **Ability to represent numerical constraints** - as seen above, MLP encompasses plenty of calculations. The ontology must be capable of representing them in an straight-forward way, as well as the constraints over them involving mathematical operations like summations, exponentiations or function derivations.
- **Availability of a rich vocabulary of abstract definitions** - the MLP ANN Ontology makes use of a lot of abstract concepts that are fulfilled either by a core mathematical ontology or by a top ontology which must include mathematical expressions, set theory, graph theory, algorithms, sequences, etc.
- **Ability to represent n-ary relations** - the freedom to use unlimited arities in relations leaves ontologists free to use some advanced constructs. For instance, a synapse connects two neurons, thus possessing a ternary relation named `connects` that has, as arguments, the synapse and the two neurons.
- **Ability to represent functions** - when dealing with mathematics concepts, we often need to represent functions. In the MLP ANN, we had to represent the activation function and the stopping criteria based on functions.

In the next section, we discuss the difficulties in fulfilling such criteria with DL.

4 Description Logics' Representational Problems regarding Mathematical Ontologies

DL and its Ontology Web Language OWL present in practice a number of limitations in expressiveness that prevented us from using it for representing the knowledge presented in the section 2. We discuss the main issues in the next subsections.

4.1 Description Logic Ontological Engagement

The first aspect to remark when embarking in such a discussion relates to the conceptual and epistemological commitment (BRACHMAN 1978) of DL as a representation formalism. Its representation purposes and consequent grounding on precise and unambiguous set theoretic semantics (BAADER ET AL, 2003) allows the language mainly to describe quite precisely domains that involve sets (as concepts or classes according to the formalism's jargon), elements (as class and relation instances), their relations (also called properties or roles) and simple constraints that can be expressed in axioms that use a limited set of constructs like relations' cardinality, type checking of relations' domain and range, classes' subsumption, complement, disjointness, equivalence, instance membership to a class and instance equality/inequality.. This is enough to make for the main reasoning task of concept classification, which means to entail class subsumption, when this relation is not declared explicitly[1]. Class subsumption, equivalence, disjointness and inconsistency (standing for a class that according to its definition cannot bear any instances) can be inferred by the reasoned, also called *classifier*.

In the next subsection we discuss the use of numerical constraints in DL.

4.2 Numerical Representations and Constraints

Despite consisting of a powerful family of semantically well-founded formalisms', DLs are not endowed with the basic mathematical resources necessary for representing neural networks, for instance the arithmetical operations. The only type of arithmetical processing is counting and comparing the number of instances that participate in relations for solving queries, like with the following axiom:

$$Worried\text{-}Woman \equiv Woman \sqcap (\geq 3\ child.Man) \qquad (6)$$

It states that a worried woman is one who has three or more male children. The classifier is able to solve queries that searches for the instances of the class *Worried-Woman* by counting the instances of *Woman* which are related to instances of the class *Man* via the relation *child*.

Although it is possible to define relations with numerical values as datatype properties in OWL (types integer and float and special types date, time and dateTime), we did not find in the literature DL extensions that address the general problem of solving arithmetical constraints, ranging from the simplest ones (like the ones that use the four basic arithmetical operations) to the most complex (using exponential, differential, integral operators, differential equations, etc).

Description Logic consists indeed in a smart purely declarative formalism that performs optimized classical reasoning. Sticking to this stance, no ultimate solution is expected to the posed problem of including arbitrary arithmetical constraints, because classifier will not cope with them, ought to the fact that elementary number theory has been proved undecidable by Gödel in his classical incompleteness theorem (GÖDEL,

[1] Classification based on subsumption is also used for solving equivalence and disjointness between classes and class inconsistency. Even in assertional queries, i.e., the ones which include instances, subsumption checking is the basic processing behind the reasoning (BAADER ET AL, 2003).

1931) as well as rational number theory, since the elementary number theory could be defined in terms of the rationals, as proved by Julia Robinson (ROBINSON, 1949).

So, the hopes for proposing a general DL solution with arithmetical constraints probably lies both on restricting the constructs for a limited set of operations and solving them with built-in constructs that should be evaluated before classification (in the flavor of Prolog, for instance). Such extension, on the other hand, would change the formalism into a more procedural and less declarative one, and is not yet reported.

4.3 The Need for a Mathematical Background Knowledge in a Top Ontology

Due to the low expressiveness of OWL, any top-ontology developed using it would face a lot of problems in representing complex abstract mathematical concepts. This statement is corroborated by the fact that, despite the existence of several efforts for the translation the top ontology Supper Upper-Merged Ontology (SUMO) to OWL, none of them proved successful. Translations can be found in the SUMO site but they are in OWL Full (which is known to be undecidable (BAADER ET AL, 2003)). Besides this fact, any translations of full first order logic to DL will just prune the knowledge from the original ontology that cannot be expressed in DL.

We indeed have tried to work with these translations, encountering many practical problems. The two more relevant for us were the loss of a the rich set of mathematical axioms provided by SUMO upper and the necessary mathematical operations and constraints that could not be defined in DL, due to the lack of expressiveness.

5 Solution: More Expressive Languages and Top Ontologies

Since DL does not fulfill our expressivity needs, the solution lies on relying on employing a more expressive language, with first-order logic (FOL) expressivity at least and, if possible, capable of defining basic arithmetic operations, and a top ontology that includes some basic mathematical definitions. A Top Ontology defines a set of generic concepts to be shared by various domain-specific ontologies. Complying with it assures semantic interoperability among the ontologies and better ontological engagement (BRACHMAN, 1978). Thus, based on the requirements enlisted above, we decided to use the SUO-KIF (Standard Upper Ontology Knowledge Interchange Format) with SUMO as the background mathematical knowledge required.

The SUO-KIF language has declarative semantics and is quite comprehensive in terms of logical expressiveness, once it was intended primarily to implement the first-order logic. It is an extension of KIF (GENESERETH, 1991), created with the objective of supporting the development of SUMO. The goal was simplifying KIF, by maintaining its syntax, logical operators and quantifiers, but leaving to the ontology itself the declarations defining classes and instances, and thus eliminating the dependency from the Frame-Ontology of KIF, what happens to Eng-Math (GRUBER & OLSEN, 1994). Instead, one must create instances based on the concepts defined in SUMO as an external resource. Another relevant capability of SUO-KIF when compared to KIF's Eng-Math is the deployment of the Sigma reasoner (PEASE, 2003). Up to now, no reasoner can take KIF's rich expressivity on. As for the top ontology, we

opted for SUMO, since it endows us with the math we needed. In the next section, we present the MLP ANN ontology, with some examples using SUO-KIF and SUMO.

2. The Resulting MLP ANN Ontology in SUO-KIF

In this section, we present the most important concepts of the Multi-Layer Perceptron Artificial Neural Network (MLP ANN) Ontology. We deploy the basic taxonomy of the ontology in Figure 2. Note that it uses plenty of concepts from SUMO, like the classes Abstract, Physical, Relation, BinaryRelation, IrreflexiveRelation, etc. Next, we take a deeper look at the ontology, describing some of the concepts and axioms that required a higher level of expressiveness.

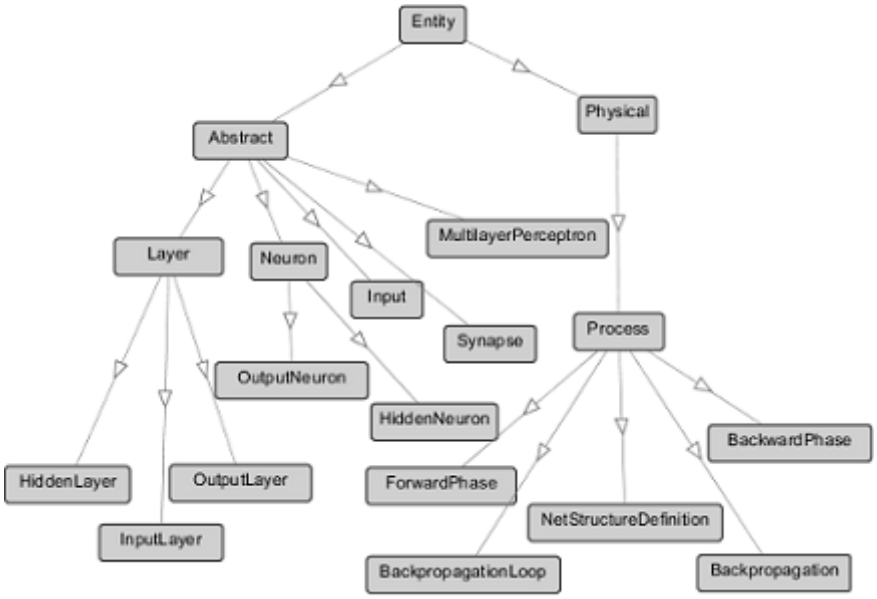


Figure 2. The ANN MLP Ontology basic taxonomy.

5.1 Activation function

The representation of the activation function constitutes a good example of a function definition that takes advantage of the SUO-KIF expressiveness and the richness of the mathematical vocabulary provided by SUMO. In the code below, we state the concept of `ActivationFunction` as a function (irreflexive, intransitive, assymetric relation) whose domain is the class Neuron.

```
(instance activationFunction BinaryPredicate)
(instance activationFunction IrreflexiveRelation)
(domain activationFunction 1 Neuron)
(domain activationFunction 2 UnaryFunction)
```

Then we represent the concept of a Logistic function, which is an unary function that receives a real number as the only argument and returns an `MLPRealNumber`.

```
(instance Logistic UnaryFunction)
(instance Logistic TotalValuedRelation)
(domain Logistic 1 RealNumber)
(range Logistic 1 RealNumber)
```

`MLPRealNumber` is defined as a real number ranging between -1 and 1:

```
(=>(instance ?NUMBER MLPRealNumber)
   (and (or (greaterThan ?NUMBER -1)
            (equal ?NUMBER -1))
        (or (lessThan ?NUMBER 1)
            (equal ?NUMBER 1))))
```

Finally we define the logistic function, as defined in subsection 2.1., def. (2):

```
(=>(equal (Logistic ?NUMBER1) ?NUMBER2)
   (equal ?NUMBER2 (DivisionFn 1
     (AdditionFn 1 (ExponentiationFn NumberE
            (SubtractionFn 0 ?NUMBER1))))))
```

Next, we describe the ANN learning process.

5.2 Backpropagation algorithm process

The process is an iteration alternating the backward and forward phases, until the halting criterion is met. We can represent it using four axioms. The first one states that the backpropagation algorithms starts defining its net structure

We can represent this process using four axioms. The first one states that for all backpropagation algorithms there exists a sub-process (the definition of the net structure) starting together with the backpropagation algorithm:

```
(=>(and(instance ?B Backpropagation)
        (instance ?NSD NetStructDefinition)
        (subProcess ?NSD ?B))
   (exists (?BL)
        (and(instance ?BL BackpropLoop)
            (subProcess ?BL ?B)
            (exactlyEarlier (WhenFN ?NSD)
 (WhenFn ?BL)))))
```

The second axiom states that right after the net structure definition, we start the iteration over the backpropagation loop. The `exactlyEarlier` relation means that the first argument finishes at the same instant that the second starts.

```
(=>(and(instance ?B Backpropagation)
        (instance ?BL Backproploop)
          (subProcess ?BL ?B)
          (not (StopCriteriaFn ?B)))
     (exists (?BL)
          (and(instance ?BL2 BackpropLoop)
            (subProcess ?BL2 ?B)
            (exactlyEarlier (WhenFN ?BL)(WhenFn?BL2)))))
```

Finally, the last axiom states that, if the halting criterion is met, the algorithm ends. The `finishes` relation means that both arguments finish at the same time:

```
(=>(and(instance ?B Backpropagation)
        (instance ?BL BackpropLoop)
          (subProcess ?BL ?B)
          (StopCriteriaFn ?B))
     (finishes (WhenFn ?B) (WhenFn ?BL)))
```

The weights are updated (subsection 2.2., definition 5) according to the axiom shown below. It states that if a weight is identified as one to be updated, then the old weight holds only until the loop cycle lasts, and the new weight is calculated and set.

```
(=>   (and
      (instance ?UPDATEWEIGHT UpdateSynapseWeight)
      (synapseToUpdate ?UPDATEWEIGHT ?SYNAPSE)
      (connects ?SYNAPSE ?NODEA ?NODEB)
      (hasOutput ?NODEB ?OUTPUT)
      (holdsDuring (BeginFn ?UPDATEWEIGHT)
      (hasWeight ?SYNAPSE ?OLDWEIGHT)))
      (holdsDuring (EndFn ?UPDATEWEIGHT)
      (and
         (hasWeight ?SYNAPSE ?NEWWEIGHT)
         (equal ?NEWWEIGHT (AdditionFn ?OLDWEIGHT
                (MultiplicationFn ?LEARNRATE
                (MultiplicationFn ?OUTPUT
                      (NeuronErrorFn ?SYNAPSE)))))))
```

5.3 An Example of Ternary Relation: The Synapse Definition

In the next encodings, we show an example of a ternary relation, the relation `connects`, that links two neurons through a synapse. The first stretch of code shows the basic characterizations of the relation in terms of inputs and outputs:

```
(instance connects TernaryPredicate)
(domain connects 1 Synapse)
(domain connects 2 Neuron)
(domain connects 3 Neuron)
```

Next, an axiom is defined, stating that all connections of a neuron are linked with a neuron from its next layer. Note that a synapse takes part of the ternary relation, thus sufficing for our representation needs.

```
(=>(and(instance ?N-A Neuron)
        (connects ?SYNAPSE ?N-A ?N-B))
   (exists(?LAYERA ?LAYERB)
          (and(instance ?LAYERB Layer)
               (instance ?LAYERA Layer)
               (neuronLayer ?N-A ?LAYERA)
               (neuronLayer ?N-B ?LAYERB)
(nextLayer ?LAYERA ?LAYERB))))
```

In the next section, we discuss related work regarding mathematical ontologies.

5.4 Discussion

The ontology defines the exact constraints that hold among the many elements of a MLP ANN, as well as the calculations and sequential operations needed to update it during the training phase. Once SUO-KIF is endowed with a reasoned able to deal with the mathematical constraints, an application that employs it – an intelligent agent, expert system, computer algebra system or theorem prover - is capable of creating a virtual MLP ANN, run it (including all the calculations and updates), and - more important in terms of declarativity – answer queries about each aspect of it, like stating how many layers at least a MLP ANN should possess, etc. With that features, even an intelligent tutor system can make a good use from its knowledge.

Note that, when an expressive formalism is used with a supportive top ontology, most, if not all, mathematical knowledge can be represented by a similar solution. Simple examples are other types of neural networks (even recurrent, constructive)

6 Related Work

In the field of Artificial Intelligence, the first successful systems to deal with mathematical contents that go beyond simple numerical calculi came from the branch known as Computer Algebra Systems (CAS) (BERTOLI ET AL 1998). They are capable of performing algebraic and symbolic computation in an abstract way A representative example of this trend was the REDUCE system (HEARN 2004), which is still in use nowadays. It is able to calculating algebraic derivates and integrals of complex functions among many other functionalities. Many of those systems employ declarative solutions for algebraic problems and some of them are able of using and presenting proofs indeed. A detailed comparison among computer algebra systems and their features can be found in [13]. Nevertheless, the integration between this type of system and the growing field of ontology is still lacking tough. The consequence is that some valuable mathematical knowledge available in mathematical ontologies, like the characterization of distinct relation types used here, are not being fully exploited in CAS systems.

As for the problem of mathematical knowledge in ontologies, we have searched the literature for possible solutions to our problems, before deciding for the SUO-KIF language. We found solutions ranging from simple syntactic agreements, like MathML (CARLISLE 2009) - a markup language without axioms -, to full-fledged languages and top ontologies like SUO-KIF and SUMO. Among them, we found a proposal to mix OWL with OpenMath to encode math contents (BUSWELL 2004). We indeed made attempts to employ this approach, by to representing the math needed in the MLP ANN ontology in OWL and processes using the Semantic Web Rule language (SWRL) built-ins. However, none of these solutions fitted our needs. All of them were limited, in the sense of depending upon external features (like the rules in SWRL), instead of using a mathematical theory. They mostly represent the expressions in a structured way and even the solutions using rules will produce fragmented pieces of knowledge that could pose problems of maintenance in the future.

Furthermore, we tried out a more consistent approach, the use of EngMath (GRUBER & OLSEN, 1994), a mathematical ontology for engineering. EngMath could indeed constitute a sound alternative, as it takes advantage of the Kif-Numbers ontology

(GRUBER & OLSEN, 1994), a KIF vocabulary focused on numbers, arithmetic operations and related definitions. We chose SUMO and SUO-KIF because, we conclude that SUMO covers all definitions comprised in Kif-Numbers and EngMath.

7 Future Work and Conclusions

Despite the popularity and usefulness of DL languages, we claim that this formalism suffers from a lack of expressiveness for the representation of mathematical knowledge. In our practical experience portrayed here, the most relevant lesson learned was that we could only overcome DL representational problems by using at least a full first-order logic formalism with a supportive top ontology like SUMO, that contains the mathematical background knowledge required to qualify and define properly the math definitions for the new ontologies. We presented a thorough case study in that direction, on the field of automatic learning, the MLP ANN ontology.

We envisage two types of future work. As for the use of the developed ontology, we are heading for practical applications of the ontology. The creation of these applications, such as an intelligent agent capable of interacting with users answering questions about artificial neural networks (making use of the ontology) are in our agenda. We also consider the translation of the ontology to other languages, such as Prolog, so as to enable us to reason with it and run concrete applications.

Another more general future work lies on the investigation of ways to embed ontologies into computer algebra systems. Devising a solution for this general problem will certainly endow the latter with more powerful reasoning techniques while solving mathematical problems. For instance, CAS systems could take advantage during inference of the applicable mathematical constraints defined as relations qualifiers in the ontology. These constraints need not be hardcoded in the systems, thus increasing knowledge reuse.

Acknowledgments. The authors would like to thank prof. Richard Banach, from the University of Manchester, England, who provided us with valuable input on the limitations of Description Logic for Mathematical representations, and prof. Jacques Calmet, from the University of Karlsruhe, Germany, who, as scientific ancestor, indirectly introduced us to the field of Computer Algebra.

References

BAADER, F., CALVANESE, D., MCGUINNESS, D.L., NARDI, D., PATEL-SCHNEIDER, P.F., eds.: **The Description Logic Handbook**. Cambridge Univ. Press (2003)

BERTOLI, P.G., CALMET, J., GIUNCHIGLIA, F., HOMANN, K.: **Specification and integration of theorem provers and computer algebra systems.** In Calmet, J., Plaza, J., eds.: AI and Symbolic Computation: Proceedings of AISC'98, Berlin, Germany, Springer (1998) 94-106

BRACHMAN, R.J.: **On the epistemological status of semantic networks.** BBN Report 3807, Bolt Beranek and Newman Inc., (apr 1978)

BUSWELL, S. CAPROTTI, O.,CARLISLE, D.,.DEWAR D., GAËTANO, M., KOHLHASE, M.: **Open math standard report**, Open Math Society (2004)

CARLISLE, D., MINER, R., ION, P.: **Mathematical markup language (MathML) version 3.0. Candidate recommendation**, W3C (dec 2009)

GENESERETH, M.R.: **Knowledge interchange format**. In Allen, J.F., Fikes, R., Sandewall, E., eds.: Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, San Mateo, California (1991) 599-600

GÖDEL, K.: **Über formal unentscheidbare sätze der principia mathematica und verwandter systeme**. Monatshefte für Mathematik und Physik 38 (1931) 173-198

GRUBER, T.R., OLSEN, G.R.: **An ontology for engineering mathematics**. In Doyle, J., Sandewall, E., Torasso, P., eds.: Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, San Francisco

HAYKIN, S.: **Neural networks**. MacMillan, New York (1994)

HEARN, A.C.: **Reduce user's and contributed packages manual, version 3.8. Report**, The RAND Corporation, Berlin, Germany (2004)

NILES, I., PEASE, A.: **Towards a standard upper ontology.** In: Proc. of the int.conference on Formal Ontology in Information Systems, New York, NY, USA, ACM (2001) 2-9

PEASE, A. **The Sigma Ontology Development Environment.** In Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems, Acapulco, Mexico, 2003

PEASE,A. **A Standard Upper Ontology Knowledge Interchange Format**. 2009

ROBINSON, J.: Definability and decision problems in arithmetic. Journal of Symbolic Logic 14 (1949) 98-114

WELTY, C., MCGUINNESS, D.L., SMITH, M.K.: **OWL web ontology language guide. W3C recommendation**, W3C (2004) http://www.w3.org/TR/2004/RECowl-guide-20040210/.

WIKIPEDIA: **Comparison of computer algebra systems**. en.wikipedia.org/wiki/Comparison of computer algebra systems, 2010.

Integration of a Domain Ontology in e-Science with a Provenance Model for Semantic Provenance Generation in the Scientific Images Analysis

Lucélia de Souza[1,2], Maria Salete Marcon Gomes Vaz[2,3]

[1]Departament of Computer Science – University of Western of Parana (UNICENTRO) Rua Camargo Varela de Sá, 03, CEP 85040-080 – Guarapuava/PR – Brazil

[2]Department of Informatics – Federal University of Parana (UFPR) Rua Cel. F. H. dos Santos, 100, CEP 81.531-980 – Curitiba/PR - Brazil

[3]Department of Informatics – State University of Ponta Grossa (UEPG) Av. Carlos Cavalcanti, 4748, CEP 84.030-900 - Ponta Grossa/PR - Brazil

{lucelias,salete}@inf.ufpr.br

Abstract. This paper describes the integration of a domain ontology in e-Science with a provenance model for semantic provenance generation in the scientific images analysis. The domain ontology is related with images obtained from the CoRoT Telescope, where the exoplanets search require detrend algorithms as preprocessing, improving the chance to detect planetary transits. In order to retrieve standardized information regarding the origin and facilitate the monitoring of information, the Proof Markup Language – PML was chosen as provenance common model due to its characteristics of modularity, reuse, interoperability and the possibility of justify how conclusions were obtained. As contribution of this paper, the integration of the ontologies presented enables getting information from the domain and justify the conclusions, through a standardized provenance model, allowing logical inference and semantic interoperability.

1. Introduction

In the scientific images analysis, information of provenance provide the source of processing, allowing share, reuse, reprocessing and do further analysis in data and process. The semantic provenance [Sahoo et al 2008] is related with the Semantic Web and can be obtained by means of ontologies [Borst 1997], which represent the knowledge, structuring information in an organized manner and generating semantic in the data.

In this work, a domain ontology was developed in the Ontology Web Language – OWL2 called CorotDataAnalysisOntology (crtdao) [de Souza et al 2011] allowing to extract domain information in the scientific images analysis. It relates with images obtained from CoRoT Telescope[1], which provides thousands of light curves in format Flexible Image Transport System – FITS [Hanisch et al 2001]. In the analysis of these images, the search of planets outside of Solar System (exoplanets) requires detrend

[1] CoRoT Archive: http://idoc-corotn2-public.ias.u-psud.fr/

and/or filter algorithms as preprocessing for removing phenomena that may occur suddenly, such as random jumps and/or trends, slow and gradual changes in certain properties of the images, under whole range of the investigation. So, different detrend and/or filter algorithms can be applied to treatment of these phenomena, improving the chance to detect planetary transits (Figure1).

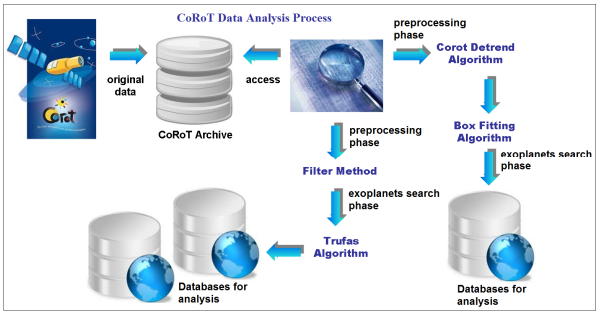


Figure1. CoRoT Data Analysis Process

However, domain ontology can be insufficient, semantically, for generation and sharing of provenance information. It is necessary to make use of a common model for the provenance generation as means to allow interoperability, reuse and extension of ontologies [McGuinness et al 2007]. In this case, provenance models can be integrated with domain ontologies because their use increases the understanding of users about how answers were generated and also facilitates the acceptance of the results. Among the provenance models existing, Provenir [Sahoo et al 2009], Open Provenance Model – OPM [Moreau et al 2011] and Proof Markup Language – PML [McGuinness et al 2007] stands out, being workflow-based systems. These models were analyzed for use in the scientific images analysis, being chosen the PML due to its characteristics of modularity, reuse, interoperability and mainly by allow us to justify how were obtained the conclusions.

The objective of this paper is to present the integration of the domain ontology with the PML Provenance Model, contributing to enrich the scientific images analysis with semantic and standardization. This integration enables getting the information from the domain and justifies the conclusions, by means of inference steps involving which the inference engine, inference rule and/or source used to generate it.

This paper is structured as follows, besides this introductory section. The second section describes about data provenance and workflows. The third section describes about domain ontologies and provenance models and the next section presents their integration. The fifth section brings the related works, followed of the conclusions and the future works.

2. Data Provenance and Workflows

Provenance means origin or source. In the scientific images analysis, provenance information proves the correctness of the resulting data, being regarded by Tan (2007) as important as the result itself.

The provenance information can have granularity in the fine-grain and coarse-grain forms [Tan 2007]. The first form involves the data derivation and storage in databases how proposed by authors Tan (2007), Buneman et al (2007), Cheney et al (2009), among others. There are two approaches, such as metadata annotation and non-

annotation approach, through queries and inverse functions used for data transformations. The second form involves activities and processes used to perform tasks of complex scientific data by mean of scientific workflows [Davidson and Freire 2008], where can be human interactions during the execution of processes flow.

2.1 Semantic Provenance in the Scientific Images Analysis

This paper stands out by enriching the data analysis semantically, related to FITS images that are available in the CoRoT Archive to exoplanets search. During the execution of detrend and/or filter algorithms, provenance information can be stored in the FITS images header.

The FITS Standard [Hanisch et al 2001] establishes rules for use of these images, which differs of the traditional format of images, due to its basic structure formed by a header containing metadata (data about data) such as SIMPLE, BITPIX, COMMENT, HISTORY, among others, and a matrix used for storing binary data.

However, the FITS specification does not contemplate the addition of provenance metadata, describing the use of HISTORY metadata to store steps executed. This form of provenance generation is free text, not being machine readable, impeding its use by software agents.

In scientific images analysis, provenance information records steps performed and generating knowledge in order to avoid reprocessing and contribute to sharing, reuse and analysis further. So, the metadata storing in images header or in the databases is insufficient, semantically, to generate provenance. This information is useful for local researchers, but not enough to share, reuse and reprocessing by scientific community. There is a need for standardization of provenance metadata to be generated and stored, just as it takes more detailed information to contemplate the real needs of researchers as to the semantic knowledge about the data generation over time. Accordingly, the next section describes the development of domain ontology in this environment.

2.2 Domain Ontology in e-Science

Ontology is defined as a formal and explicit specification of a shared conceptualization [Borst 1997]. It is characterized as a mean of representing knowledge, structuring information in an organized manner of a domain and generating semantics in the data.

The development process of the domain ontology proposed is based on Ontology Development 101 [Noy and McGuinness 2001]. We started by identifying a set of competency questions from domain that must be answered by the ontology, such as: *What are the statistical techniques (Linear, Polynomial, among others) used by detrend algorithms?; The CoRoT Detrend Algorithm treats which systematic effects?; What transit algorithm had the type of method Least-Squares?;* among others. From these questions, were identified classes, their relationships and the instances. Restrictions are declared using axioms and/or rules, providing semantics and allowing inferences.

The Protégé 4.1 tool [Knublauch et al 2004] was used to the development of the domain ontology and the generation of knowledge base. The used language is OWL 2.0,

recommend by World Wide Web Consortium - W3C and based on Descriptive Logic – DL [Baader 2003]. Pellet 2.2 is used to verify its consistency.

An OWL ontology in abstract syntax contains annotations, axioms and facts. However, the use only of axioms presents expressive limitations, mainly with the use of properties such as composition of roles [Horrocks et al 2005].

The composition of roles as '*isAlgorithmDetrendPolynomialOf*' shows an example. If an algorithm is *AlgorithmDetrend* and the method type is *Polynomial*, then the algorithm is an *AlgoritmDetrend* of the *Polynomial* type. The relationship between the composition of the '*isAlgorithmDetrendOf*' and '*isMethodTypePolynomialOf*' properties and the '*isAlgorithmDetrendPolynomialOf*' property is limited to the form $P \circ Q \sqsubseteq P$, in order to maintain decidability. The composition of two properties is a subproperty of one of the composed properties, that is, the complex relationship between composed properties cannot be captured. This is the case of '*isAlgorithmDetrendPolynomialOf*' property that cannot be captured because it is not one of '*isAlgorithmDetrendOf*' not '*isMethodTypePolynomialOf*'.

So, the complex axiom '*isAlgorithmDetrendOf*' $\circ$ '*isMethodTypePolynomialOf*' $\sqsubseteq$ '*isAlgorithmDetrendPolynomialOf*' presents the form $R \circ S \sqsubseteq T$ and $T \circ S \sqsubseteq R$, because exists cyclical dependences in the definition, violating the irreflexivity. This verification is important in relation of the decidability, because such cyclical dependences can induce undecidibility and the use in an ontology should be restricted. One way to address this problem is extend OWL with a more powerful language to describe properties.

Horrocks et al (2005) extends axioms OWL DL to allow rule axioms (a Semantic Web Rule Language - SWRL), in the form: *axiom::=rule*. In the human readable syntax, a rule has the form antecedent$\rightarrow$consequent, an implication between an antecedent (body) and consequent (head).

Informally, a rule means "if the antecedent hold (is true), then consequent must also hold". The antecedent and consequent of a rule consist of zero or more atoms, which can be of the form *C(x)*, *P(x,y)*, *sameAs(x,y)* or *differentFrom(x,y)*, where *C* is an OWL DL description, *P* is an OWL property, *x* and *y* are either variables, individuals or data values. Multiple atoms in an antecedent are treated as conjunction and multiple atoms in a consequent are treated as separate consequences [Horrocks et al 2005].

The Protégé 4.1 Tool allows working with rules from View Rules. Pellet supports reasoning with SWRL rules, which interprets SWRL using the DL-Safe Rules notion, where rules will be applied only to named individuals in the ontology.

2.3 Analysis of the domain ontology as to semantic integration

The domain ontology was evaluated by domain experts as the terms used as well by ontologists. Also was formalized in an extension of the DL called *SROIQ* [Horrocks et al 2006], which presents characteristics of the expressiveness, decidability and robust computational properties, being an extension more expressive than the Attributive Language, the most basic family of DL.

The formalization allow us to specify the ontology independently of the domain, contributing to verification and validation of axioms assertional, terminological and role

inclusion used, well as allows to infer knowledge. With the formalization in *SROIQ* DL, under OWL 2, recommended in 2009 by the W3C, also is possible to verify the consistency of the knowledge base. In this way, it is feasible to enrich the scientific images analysis with semantic and standardization.

The domain ontology proposed in [de Souza 2011] presents as main classes: *DataSet*, *Methods*, *Technique*, *AlgorithmBase*, *PeriodicSignalShape*, *MethodType*, *Algorithm*, *Software*, *Metadata*, *Person*, *Run*, *Telescope*, *Language* and *SistematicEffectType*. *Header* class was created to relate header specific metadata of FITS images and the *Database* class, related with the storage location. The Language class was specified in *ProgramationLanguage*. However, aiming semantic integration in e-Science, a domain ontology developed was evaluated in relation to existing ontologies (Figure2).

The VSTO ontology[2] stands out as an ontology open-source, extensible and reusable in the area of solar-terrestrial physics, which supports interdisciplinary projects of virtual data collections. This ontology was analyzed, and made the following adjustments in the domain ontology: i. *Telescope* class was inserted as a subclass of *Instrument* and were also imported from VSTO the following classes: *DataProduct* related with FITS images, which were previously represented as *DataSet*; *vsto:InstrumentOperationMode* related with information about the operation mode of the instrument; *vsto*:*DateTimeInterval*, being intervals for date and time and *vsto:Parameter*, including the following parameters: *ErrorParameter*, *Noise*, *Period*, *SignalToNoiseRatio*, *TimeDependentParameter* and *StatisticalMeasure*.

The Semantic Web Earth and Environmental Terminology - SWEET Ontology[3] has widespread acceptance in e-Science. However, this ontology extends more in width than depth in certain areas. Thus, for purposes of interoperability and reuse, the *crtdao:MethodType* class was replaced by import of the *sweet:Process* class. It's because the objective of this work is to deepen concepts to generate semantic provenance as the statistical methods used in the analysis of FITS images.

3. Domain Ontologies and Provenance Models

Domain ontologies should be built based on Foundation Ontology, such as SUMO[4], DOLCE[5], UFO[6], among others, because they are theoretically well-founded, becoming the category systems independent of domain, describing the general concepts and improving the quality of conceptual model [Guizzardi 2005]. They are characterized by being highly reusable because it shapes basic and general concepts, as well as relations. However, the well-founded ontologies are generic about many areas.

So, due to the need for representing provenance information, provenance models stands out because are ontologically well-founded representation models, adding concepts and relationships provenance-aware, allowing the adoption of a common provenance terminology [McGuinness et al 2007]. These models are presented follow.

---

[2] *Virtual Solar-Territorial Observatory:* http://escience.rpi.edu/ontology/vsto/2/0/vsto.owl
[3] *Semantic Web Earth and Environmental Terminology*: http://sweet.jpl.nasa.gov/
[4] *Suggested Upper Merged Ontology*: http://www.ontologyportal.org/
[5] *Descriptive Ontology for Linguistic and Cognitive Engineering* http://www.loa.istc.cnr.it/DOLCE.html
[6] *Unified Foundational Ontology*: http://code.google.com/p/ufo-nemo-project/

3.1 Open Provenance Model - OPM

It is an abstract model developed from Provenance Challenge Series to explain how artifacts were derived, based on workflows. It is independent of technology for interoperability purposes. Uses a graph based on a syntactic rules set and topological constraints. It presents as concepts *Agent*, denoting people; *Process*, denoting actions or executions of process; and *Artifacts*, denoting the entity produced or manipulated. This data model has applicability mainly in biologic area.

The modularity of this data model involves OPM Specification, OPMV Vocabulary, OPMO Ontology and XML Schema. The focus is on provenance in workflows, defining a small set of key concepts to general entities and relationships (*wasGeneratedBy* - WGB and *WasControledBy* - WCB) in workflows. On the downside, the OWL Profile is still evolving to adapt the OPM Specification.

3.2 Provenir

This ontology presents as main concepts *Agent*, *Process* and *Data*. *Data_Collection* and *Parameters* spatial, domain and temporal are subclass of the Data. It is constituted by eight classes and eleven properties, including the Relation Ontology.

It presents as characteristics a common model to represent provenance, being expressive as the concepts and relationships modeled named well-defined, can be extended to modeling of complex provenance information and domain-specific, enabling analysis in SWRL and W3C Rule Interchange Format - RIF. This ontology has applicability in biomedical and oceanography areas in real projects of the e-Science.

3.3 Provenance Markup Language - PML

PML is based on Proof Theory and constitutes a common model for represent and share explanations generated by various intelligent systems such as answers systems of hybrid web questions, analytical text, theorem provers, among others. It describes the justifications as a sequence of information manipulations steps used to generate a response. This sequence is referred as a proof.

Due to modularity, it is possible to use modules individually for *Provenance* (PML-P), *Justification* (PML-J) or hold *Trust* (PML-T) in the data[7]. The PML-T supports annotation of complex trust relations in provenance concepts and justifications. The primitive concepts and relations are specified in OWL, facilitating reuse and extension. The modules PML-P and PML-J are described in the following.

3.3.1 Provenance Ontology - PML-P

PML provides a vocabulary for justification of metadata whose focus is on representational primitives used to describe properties of 'things' identified as information, language and resources, such as organization, person, agent and services. These primitives are extensible, used to annotate the source of information, as to represent sources used and who encoded the information. PML-P presents the following concepts.

[7] URL: http://inference-web.org/2007/primer

An instance of *IdentifiedThing* refers to a real world entity and its properties note the properties of entities such as name, description, date and time of creation and ownership. PML-P also includes *Information, Source and SourceUsage, Language* and *InferenceRule* subclasses.

The *Information* subclass supports references to information on various levels of granularity and structure, such as a formula in a logical language, a fragment of natural language or a dataset. The *Source* is extensible and refers to a container of information, such as a Document, an Agent, among others. *SourceUsage* is used to associate *Information* and *Source*, declaring information from a *Source* at certain time. *Language* represents the language in that the conclusion is represented. *InferenceRule* aims to encode various types of computation steps.

3.3.2 Justification Ontology - PML-J

This module requires concepts to represent conclusions, zero or more sets of antecedents of the conclusion and the steps used to manipulate information to get conclusions from the set of antecedents and so on recursively. The vocabulary for explanations of data focuses on representational primitives used to explain dependencies between 'things', including constructors to represent how conclusions are derived. It presents the *NodeSet* and *InferenceStep* concepts.

The *NodeSet* represents a conclusion and a set of alternative steps, each of which may provide an alternative justification for a conclusion. This term captures the concept of a set of nodes in steps from one or more proof trees deriving the same conclusion.

An *InferenceStep* represents a justification for the conclusion of the respective *NodeSet*. It refers to a logical step of inference, an information extraction step, any step in the process of computing, or an assertion of a fact or an assumption. It can also be a complex process as web service or application. An *InferenceStep* represents the details such as the *InferenceEngine, InferenceRule,* and the set of antecedents *NodeSets* of one justification for the conclusion of the corresponding *NodeSet*.

4. Integration of the Domain Ontology with the Provenance Model PML

In this work, we choose to make use of the PML-P and PML-J modules of PML model, mainly because allows us represent and explain how the conclusions were obtained by informing which the inference engine, the rules and the source of information used, as well as due to modular design.

The integration (Figure2) is done using multiple inheritance of the classes as in Zednik el al (2009), where an individual is defined as a type from the provenance model and at least one type from the domain ontology, e.g. the CoRoT instance is defined as belonging to classes *crtdao:Telescope* and the extension *pmlp:Telescope*, being an subclass of *pmlp:Agent* of the *pmpl:Source* class. So, an instance of *crtdao:Telescope* becomes the source used to justify a conclusion from a *NodeSet*. The same classes in *crtdao* e *pmlp* are treated as equivalent classes.

From a *Question* is created the respective *Query*, which is linked to a *NodeSet*, where is stated a conclusion (*Information*) through the *hasConclusion* property. *NodeSet* may have none, one or more *InferenceSteps* stating which *InferenceRule, InferenceEngine* and/or *Source* were used, beyond of a list of antecedents *NodeSets.*

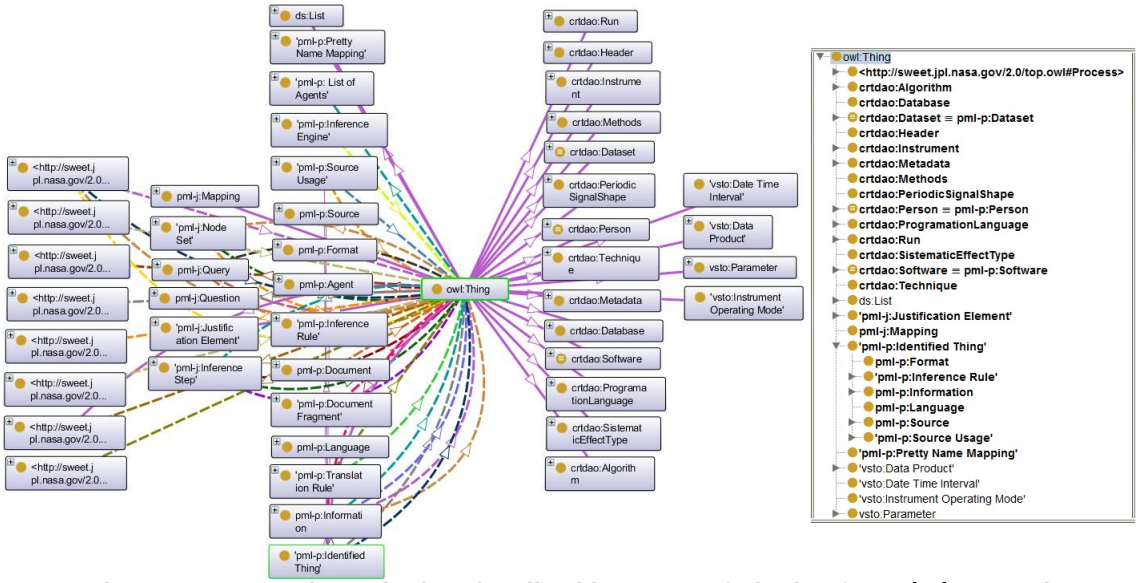


Figure2. Integrated Ontologies visualized in OntoGraf Plugin of Protégé 4.1 Tool

Given the Question '*What is the source of a given dataset*?' is specified the *Query* in a given language binds to a *NodeSet* and declares as conclusion the respective source. As inference step, it is possible to declare the *Source* of the using the property *hasSource*. So, it is possible to justify the source of information in the standardized way.

McGuinness et al (2007) identifies four types of justifications for a given conclusion, exemplified below in XML format and Protégé 4.1 Tool:

i. *The conclusion is an unproven conclusion or goal*. No justification is available and none InferenceStep is associated with the NodeSet. For the Question *What is the technique of the Photometric Detrend Algorithm?* and the Query in the Manchester OWL DL Query syntax (Figure3), the conclusion is given by NodeSet respective using the properties *pmlp:hasLanguage* related with the language of the conclusion and *pmlp:hasRawString* related with the content of information as a string.

<pmlj:NodeSet
nsPhotometricDetrendingAlgorithmTechnique>
<pmlj:hasConclusion>
<pmlp:Information>
<pmlp:hasLanguage>(English)>
<pmlp:hasRawString datatype="string">
(Photometric_Detrending_Algorithm
hasTechnique value Photometric)
</pmlp:hasRawString>
</pmlp:Information>
</pmlj:hasConclusion>
</pmlj:NodeSet>

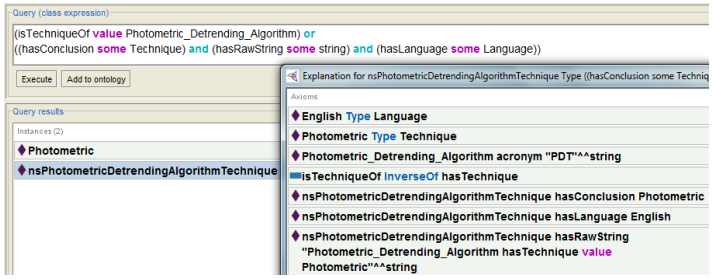


Figure3. Justifying a unproven Conclusion without InferenceStep

ii. *The conclusion is an assumption*. The conclusion is directly assumed by an agent as a true statement. The Question *What Methods the SARS algorithm belongs to?* is justified by inference in the NodeSet respective that includes the information *pmlp:hasRawString* and *pmlp:hasLanguage*. As a consequence of the InferenceStep is declared *assumption* as InferenceRule and *Pellet* as InferenceEngine (Figure4).

<pmlj:NodeSet nsSarsAlgorithmMethods>
<pmlj:hasConclusion>
<pmlp:Information>
<pmlp:hasRawString datatype="string">
(SARS_Algorithm hasMethods value
Data_Analysis)
</pmlp:hasRawString>
<pmlp:hasLanguage>(English)>
</pmlp:Information>
</pmlj:hasConclusion>
<pmlj:isConsequentOf>
<pmlj:InferenceStep>
<pmlj:hasInferenceEngine>Pellet
<pmlj:hasInferenceRule>assumption
</pmlj:InferenceStep>
</pmlj:isConsequentOf>
</pmlj:NodeSet>

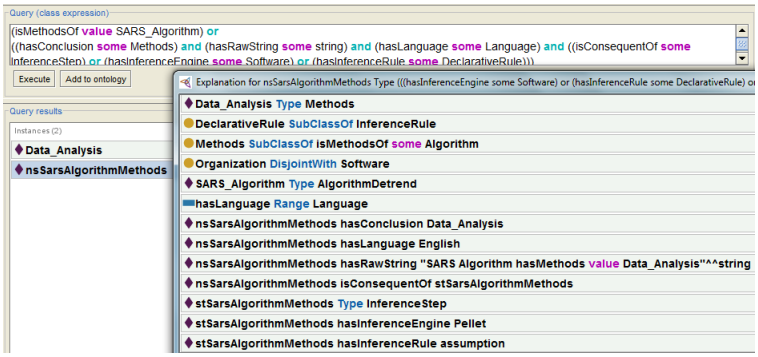


Figure4. Justifying a Conclusion using InferenceStep

iii. *The conclusion is a direct assertion*. It can be declared by Inference Engine directly without using any antecedent information (Figure5). For the Question *What is the publication of Corot Detrend Algorithm?*, the *NodeSet* respective declare the information using *pmlp:hasRawString* and *pmlp:hasLanguage* properties. As the consequence, the *InferenceStep* informs *direct_assertion* using *pmlj:hasInferenceRule* and the *pmlp:hasDocument* property informs the *Source*. Also it is possible to declare other details about the publication how number of pages and URL.

<pmlj:NodeSet nsCorotDetrendAlgorithmPublication>
<pmlj:hasConclusion>
<pmlp:Information>
<pmlp:hasRawString datatype="string">
(Corot Detrend Algorithm hasPublication value An algorithm for
correction CoRoT raw light curves)
</pmlp:hasRawString>
<pmlp:hasLanguage>(English)>
</pmlp:Information>
</pmlj:hasConclusion>
<pmlj:isConsequentOf>
<pmlj:InferenceStep>
<pmlj:hasInferenceRule>direct_assertion
<pmlp:SourceUsage>
<pmlp:hasDocument>An algorithm for correction…
<pmlp:hasFromOffset>1</pmlp:hasFromOffset>
<pmlp:hasToOffset>8</pmlp:hasFromOffset>
…

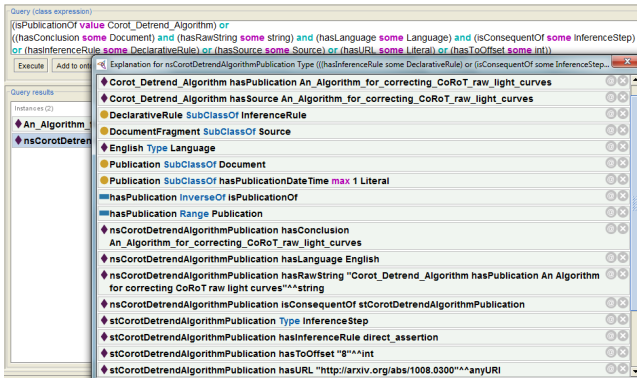


Figure5. Justifying a Conclusion using Direct Assertion

iv. *The conclusion is derived from a list of antecedents by applying a certain computation*. This representation to encode many types of computation steps. The Question *Which is the function type of the Corot Detrend Algorithm?* (Figure6) shows that the conclusion is derived from first NodeSet or from rest NodeSet.

<pmlj:NodeSet nsCorotDetrendAlgorithmFunction>
<pmlj:hasConclusion>
<pmlp:Information>
<pmlp:hasLanguage>(English)>
<pmlp:hasRawString>Corot_Detrend_Algorithm
hasFunction polynomial
</pmlp:Information>
</pmlj:hasConclusion>
<pmlj:isConsequentOf>
<pmlj:InferenceStep>
<pmlj:hasAntecedentList>
<pmlj:NodeSetList>
<ds:first>nsCorotDetrendAlgoritmPublication
<pmlj:hasIndex>0</hasIndex>
</ds:rest>nsDetrendpolynomial
<pmlj:hasIndex>1</hasIndex>
…

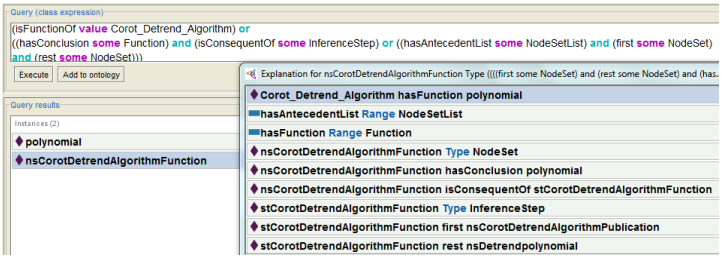


Figure6. Justifying a Conclusion derived from AntecedentList

5. Related work

Zednik et al (2009) present how the semantic provenance is reconstructed to data products in coronal physics area. This work provides a foundation for scientific workflow provenance applications, describing the use of semantic web technologies to encode provenance and domain information and demonstrating how both can be used together to satisfy complex use case. The data model use OWL ontologies independent. The Solar-Terrestrial Ontology – VSTO is used as a core domain model in e-Science, modeling data products, instruments and parameters. The provenance model uses the Inference Web and the Framework PML, chosen because of its capacibilities of represent conclusions, justificatives and explanations. The integration of provenance and domain models is done by means of multiple-inheritance from individuals´ declarations of the ontologies. The search results can be seen by Inference Web browser or by Probe-It!, enabling scientists to better understand imperfections and processing consequences upon e-Science data images.

Malaverri et al (2012) presents an approach of provenance to ensure the quality of geospatial data, combining features provided by the OPM and FGDC geographic metadata standards. It presents a case study in agriculture area, considering the trustworthiness of source, is that, the degree of confidence of who created/made available the data and temporality dimensions including valid and transaction time, e.g. 'when' related to data quality. Despite the proposal model be based on OPM model, is added own characteristics taking into account the geospatial domain and assessment of data quality. As future works, techniques to compute and assess the trustworthiness of data will be investigated.

Salayandia et al (2012) propose a framework to support the creation of ontologies for management of scientific data, specifying an abstraction in the form of a top-level ontology codified in OWL-DL, including general concepts that can be specialized to describe the capture and transformation of data. The Ontology Driven Workflow (WDO.owl) is proposed and presents three basic concepts: Date, things that can be used directly or indirectly as evidence, e.g. the output of a sensor; Method, things that can be used to transform the data, e.g. visualization of software; and Container, things that can be used as acquires or placeholders of the data, e.g. a database. WDO is specified in Description Logic and the knowledge representation system is divided into Tbox terminology and Abox, including assertions as the individuals in relation to the Tbox. WDO is aligned with PML, where the concepts Date, Container and Method are included respectively by PML concepts: Information, Source and Inference Rule. The formalism that aligns the WDO and PML Ontologies is also specified using DL, including subsumption equations rather than equalities due to concepts related with the provenance are more general than the concepts of the WDO Ontology. It is because data can be transformed by systematic processes, where the framework can be used to document the process.

This paper stands out by enrich with semantic and standardization the phases of the detrending and exoplanets search, providing information about the semantic provenance of data and statistical methods used in the correction and analysis of FITS images, contributing for adding semantic knowledge in experiments of e-Science and take advantage of the features provided by PML.

6. Conclusions

It is presented in this paper that is possible to generate semantic provenance in scientific images analysis. The environment involves FITS images from CoRoT Archive and the integration of data models related to domain ontology and the provenance model. This integration allow us make use of a common model and standardized for generating provenance, contributing for semantic interoperability and allowing us to justify how conclusions were obtained in the knowledge base.

Due to need for representing provenance information, provenance models are ontologically well-founded, adding concepts and relationships provenance-aware, allowing the adoption of a common provenance terminology. In this work, we choose to use the PML model by allowing us to represent and explain how the conclusions were obtained providing the inference engine, the inference rules and the source used, as well as due to its modularity.

The semantic provenance information obtained will be persisted in databases, and integrated in a web framework, facilitating the information retrieval processes, where queries of provenance can be performed, allowing further analysis and contributing to enrich semantically the development of scientific experiments. Despite the scope of this work, results can be expanded to fields of e-Science where the scientific images analysis requires preprocessing, adding semantic knowledge and allowing interoperability.

Acknowledgments

We acknowledge the support of the Astronomical Observatory AstroUEPG.

References

Baader, F. and Calvanese, D. and McGuinness, D. L. and Nardi, D. and Patel-Schneider, P. F. (2003). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, USA.

Borst, W. N. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Doctoral Thesis. University of Tweenty, Centre for Telematica and Information Technology, Enschede, The Netherlands, (227 pp.).

Buneman, P. and Khanna, S. and Tan, W. C. (2007). Why and Where: A Characterization of Data Provenance. In ICDT (pp. 316–330).

Cheney, J. and Chiticariu, L. and Tan, W. C. (2009). Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases (Vol. 1, N. 4, pp. 379–474). Hanover, MA, USA: Now Publishers Inc.

Davidson, S. B. and Freire, J. (2008). Provenance and Scientific Workflows: Challenges and Opportunities. In Proceedings of ACM SIGMOD (pp. 1345–1350).

De Souza, L. and Vaz, M. S. M. G. and Emílio, M. and da Rocha, J. C. F. and Boufleur, R. (2011). Data Analysis Provenance: Use Case for Exoplanet Search in CoRoT Database. Presented In: ADASS XXI Conference Series. In Press: ASP Conf. Ser. Vol. TBD. San Francisco. 2012.

Guizzardi, G. (2005) Ontological Foundations for Structural Conceptual Models. PhD Thesis (CUM LAUDE), University of Twente, The Netherlands. Published as the book Ontological Foundations for Structural Conceptual Models, Telematica Instituut Fundamental Research Series No. 15.

Hanisch, R. J. and Farris, A. and Greisen, E. W. and Pence, W. D. and Schlesinger, B. M. and Teuben, P. J. and Thompson, R. W. and Warnock III, A. (2001). Definition of the Flexible Image Transport System (FITS). A&A (Vol. 376, pp. 359–380).

Horrocks, I. and Kutz, O. and Sattler, U. (2006). The Even More Irresistible SROIQ. Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), (pp. 57–67), AAAI Press.

Horrocks, I. and Patel-Schneider, P. F. and Bechhofer, S. (2005) OWL Rules: A Proposal and Prototype Implementation. Journal of Web Semantics, V. 3, N. 1.

Knublauch, H. and Fergerson, R. W. and Noy, N. F. and Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. 3rd ISWC 2004, Hiroshima, Japan. (pp. 229–243).

Malaverri, J. E. G. and Medeiros, C. B. and Lamparelli, R. C. (2012) A provenance Approach to Assess Quality of Geoespatial Data. Symposium on Applied Computing.

McGuinness, D. L. and Ding, L. and da Silva, P. P. and Chang, C. (2007). PML 2: A Modular Explanation Interlingua. In Proc. of the AAAI 2007 Workshop on Explanation-aware Computing (pp. 22–23).

Moreau, L. and Clifford, B. and Freire, J. and Futrelle, J. and Gil, Y. and Groth, P. and Kwasnikowska, N. and Miles, S. and Missier, P. and Myers, J. and Plale, B. and Simmhan, Y. and Stephan, E. and den Bussche, J. V. (2011). The Open Provenance Model Core Specification (v1.1). Future Generation Computer Systems (Vol. 27, N. 6, pp. 743–756). Amsterdam, The Netherlands: Elsevier Science Publishers B. V.

Noy, N. F. and McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Tech. Rep. KSL-01-05 and Stanford Medical Informatics Tech. Rep. SMI-2001-0880 (25 pp.).

Sahoo, S. S. and Sheth, A. and Henson, C. (2008). Semantic Provenance for eScience. IEEE Computer Society, pp. 46-54.

Sahoo, S. S. and Weatherly, D. B. and Mutharaju, R. and Anantharam, P. and Sheth, A. and Tarleton, R. L. (2009). Ontology-Driven Provenance Management in eScience: An Application in Parasite Research. Proc. of the CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II. OTM '09. (pp. 992–1009). Berlin, Heidelberg: Springer-Verlag.

Salayandia, L. and Pinheiro, P. and Gates, A. Q. (2012). A Framework to Create Ontologies for Scientific Data Management. University of Texas at El Paso.

Tan, W. C. (2007). Provenance in Databases: Past, Current, and Future. IEEE Data Eng. Bull. (Vol. 30, N. 4, pp. 3-12).

Zednik, S. and Fox, P. and McGuinness, D. L. and da Silva, P. P. and Chang, C. (2009). Semantic Provenance for Science Data Products: Application to Image Data Processing. First International Workshop on the role of Semantic Web in Provenance Management.

An Evaluation of Annotation Tools for Biomedical Texts

Kele T. Belloze[1], Daniel Igor S. B. Monteiro[2], Túlio F. Lima[2], Floriano P. Silva-Jr[1], Maria Cláudia Cavalcanti[2]

[1]Laboratório de Bioquímica de Proteínas e Peptídeos – Instituto Oswaldo Cruz
Avenida Brasil 4365 – 21.040-360 – Rio de Janeiro – RJ – Brazil

[2]Departamento de Ciência da Computação
Instituto Militar de Engenharia (IME) – Rio de Janeiro, RJ – Brazil

{kele,floriano}@ioc.fiocruz.br,{daniel_igor18, tulioflima}@hotmail.com,
yoko@ime.eb.br

Abstract. *Biomedical texts are a rich information source that cannot be ignored. There are several text annotation tools that may be used to extract useful information from these texts. However, the multi-domain characteristic of these texts, and the diversity of ontologies available in this area, demands a careful analysis before choosing an annotation tool. This work presents an evaluation of the existing annotation tools, with focus on biomedical texts. Initially, based on a set of required characteristics, a tool selection was conducted. AutôMeta and Gate tools were selected for a more detailed evaluation. They were quantitatively and qualitatively evaluated. Results of such evaluation are discussed and bring to light the best/worst of each tool.*

1. Introdução

The constant growth of data and publications in the Biomedical area has been pushing the creation and reuse of domain ontologies in that area, not only for structured data annotation, but also for text indexation and annotation. Particularly, text bases are a rich information extraction source, since many biomedical findings are available only in textual format. PubMed[1] is one of the most popular digital biomedical citation reference (more than 21 million texts). Each text citation is associated (indexed) using MeSH[2] thesaurus. However, in order to facilitate the extraction of information from texts, a more automated and detailed indexation is required.

Biomedical area texts are typically multi-domain, and require different ontologies for their annotation. The Open Biological and Biomedical Ontologies (OBO) Foundry [Smith et al. 2007] and the NCBO BioPortal [Noy et al. 2009] provide together more than 300 ontologies. The motivation of this work is to provide support for annotation with multiple ontologies. For instance, a paper about drug targets usually refers to proteins, diseases, organisms, pharmacogenomics, etc. Each of these terms can be annotated by different domain ontologies such as: GO (Gene Ontology) [The Gene Ontology Consortium 2000], for gene and protein annotations, NCBITaxon[3] (NCBI organismal classification), for organisms, and PHARE (The PHArmacogenomic

[1] http://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.nlm.nih.gov/mesh/
[3] http://bioportal.bioontology.org/ontologies/1132/

RElationships Ontology)[4], for pharmacogenomics techniques, such as the knockout technique. Based on these annotations, it is possible to establish useful correlations. For instance, a text may describe that the application of the knockout technique over a certain gene G of an organism O, led to its death. Thus, if annotated with the mentioned ontologies, an additional annotation extracted from this text would inferred: gene G is essential for organism O.

There are already a variety of (semi) automatic tools for text annotation, i.e., which provide support for the association of text expressions to ontology terms. The main goal of this work was to identify and compare such tools, with focus on texts and ontologies of the biomedical area. Initially, a set of tools has been selected. After, relevant requirements for biomedical text annotation, such as the support for manual and automatic annotation, and the flexibility for loading ontologies were observed. Only two of the selected tools attended these requirements. These two tools were then analyzed with respect to their annotation results, in terms of quantity and quality. An additional contribution of this work is to provide guidelines for annotation tool analysis.

The remainder of this work is organized as follows: section 2 introduces semantic annotation basic concepts and illustrates it in the biomedical scenario. Section 3 describes and analyzes semantic annotation tools. Section 4 reports the realized experiment, results and difficulties. Finally, conclusions and future works are presented in Section 5.

2. Semantic Annotation

Semantic annotation is an approach to achieve the concepts of the Semantic Web, whose information organization provides a means, in which the logical connection of terms establishes interoperability between systems [Shadbolt et al. 2006]. It proposes to annotate a document using semantic information from domain ontologies. Popov et al. (2003a) define semantic annotation as a "specific schema for generation and use of metadata, enabling new methods of information access". According to Ding et al. (2006), the semantic annotation should be explicit, formal and unambiguous, so that is publicly accessible, understood and identifiable, respectively.

More specifically, we emphasize that semantic annotation is an association between relevant expressions or terms of a document or from metadata, and concepts and instances described in the ontology. Figure 1 illustrates the associations between terms in a piece of text and terms of ontologies and taxonomies, and how these associations can enrich the text with the knowledge embedded in the ontology. Annotations can be inserted in the same document file or saved separately. They contribute to the information retrieval mechanisms that are able to interpret them.

The multi-domain characteristic of biomedical articles makes it difficult to obtain a well-annotated text with a single ontology. The ontologies of this area are built focused in only one domain. Therefore, for an article to be well-annotated, the use of multiple ontologies or taxonomies is needed. However, as mentioned previously, there are many available ontologies. Hence, a prior analysis of which ontologies are compliant with the domains of the articles is needed. In Figure 1 we can see that in a

[4] http://bioportal.bioontology.org/ontologies/1550/

small text fragment it was necessary to use a thesaurus, a taxonomy and an ontology for the annotation.

3. Technologies to Support the Semantic Annotation of Texts

There are tools that provide support for the semantic annotation of documents (or texts) available on the Web. These may include different characteristics such as access to ontologies, intuitive graphical user interface, editors and repositories for ontologies storage, etc.

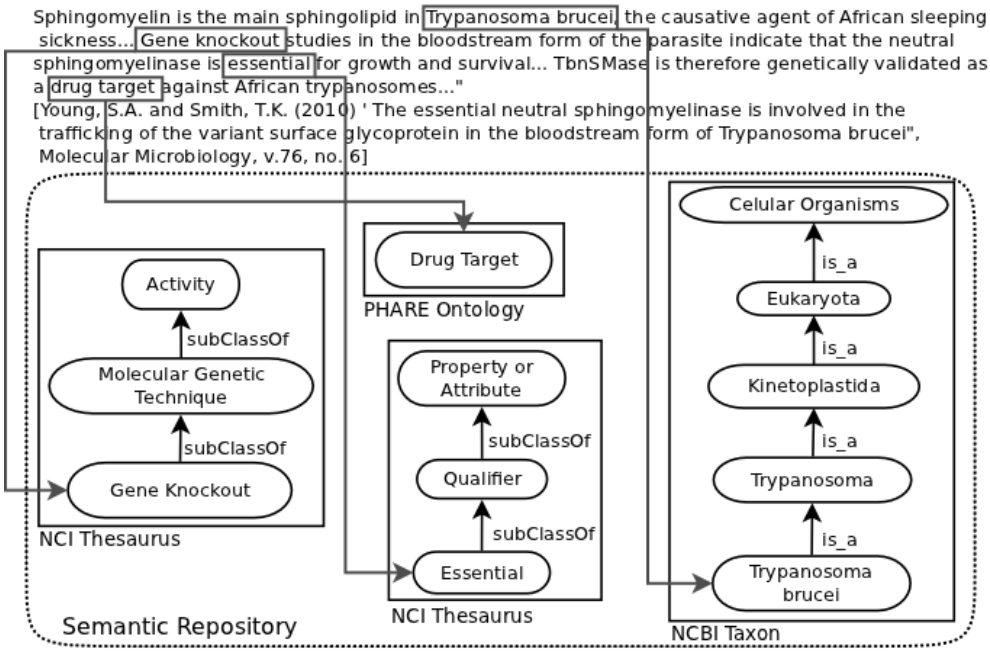


Figure 1. Associations between term_article and classe_ontology for ontology-based semantic annotation.

Regarding the kind of annotation, they are classified as manual and automatic. In the manual annotation, the user performs the whole process of marking the document, selecting the parts to be annotated and describing the annotation associated to a term of an ontology. In automatic annotation, the tool performs the annotation without user intervention, through the use of techniques such as natural language processing (NLP), machine learning and information extraction among others, to associate text expressions to ontology terms. There are tools that provide support for both manual and automatic annotation, and are considered to be hybrid. Another important characteristic is how the annotation is saved. It can be intrusive, which means the annotation is saved in the document, or non-intrusive, which means the annotation is stored in another file and do not modify the original document.

Other characteristics refer to the types of software platforms (desktop and Web), types of documents to be annotated (txt, pdf, etc.) and the use of ontologies for the annotation (which format and whether a user-choice ontology may be used). For this work, several tools have been analyzed and tested in accordance to these characteristics and are presented in the following section.

3.1. Semantic Annotation Tools

The selected tools are presented below, and Table 1 summarizes the characteristics previously described, observed in each of these tools.

Annotea [Kahan et al. 2001] is a project of the World Wide Web Consortium (W3C). The annotations of this tools refer to comments, notes, explanations or general comments on Web documents. It is part of the efforts of the Semantic Web and uses an annotation scheme based on Resource Description Framework (RDF). It stores the metadata of the annotations locally or on annotation servers.

Annozilla[5] has the same characteristics as Annotea, but works as a plugin for Mozilla Firefox browser. It stores the annotations as RDF on a server. It also highlights the annotation of the documents, which remains when it is reloaded.

AutôMeta (Automatic Metadata annotation tool) [Fontes 2011] allows the annotation of one or more documents using an ontology previously selected. The annotations generated by the tool are stored using RDFa standard (Resource Description Framework in attributes)[6].

GATE (General Architecture for Text Engineering) [Cunningham et al. 2002] is a tool for natural language processing applications. It integrates a development environment which includes plugins and other components that allow both the annotation or information extraction.

GoNTogle [Bikakis et al. 2010] is a tool for annotation and search. It also provides search facilities using a combination of semantic search and keyword. The annotations are saved as an instance in the ontology server and added to a list on annotations editor.

KIM [Popov et al. 2003b] is a web based platform for semantic search and annotation of data and documents. It has its own ontologies which includes general interest entities. The access to the features at KIM server is done through a Web interface (KIM Web UI), which allows traditional methods of searching by keyword or semantic search (entities, patterns).

Knowtator [Ogren 2006] is a plugin of Protégé, and allows an increase in ontologies to adapt to the user application. The annotation is done with the ontologies present in Protégé, from the region of the text selected to be annotated and the specification of the ontology to be used.

Melita [Ciravegna et al. 2002] is a tool that has its own ontologies, allowing the users to add their results to the used ontology, increasing it in each satisfactory annotation.

MnM [Vargas-Vera et al. 2002] is a tool that allows annotation on Web pages. It uses a learning algorithm on the annotations to posteriorly calculate the precision and recall of the annotations in the corpus. It integrates a Web browser with an ontology editor and provides APIs (Application Programming Interface) for connection between ontologies servers and information extraction tools.

[5] http://annozilla.mozdev.org/
[6] http://www.w3.org/TR/xhtml-rdfa-primer/

ONTEA [Laclavik et al. 2006] uses its own ontologies which is only related to addresses, names and e-mails.

RDFaCE (RDFa Content Editor) [Khalili and Auer 2011] is a plugin for TinyMCE Javascript WYSIWYG Editor that allows the intrusive annotation in RDFa standard. Instead of ontologies, uses APIs that suggest resources for the annotation. These resources provide appropriate URIs for objects, properties and namespaces.

RDFa Editor [Duma 2011] presents itself as a promising tool that uses RDFa as standard for the annotations. It allows arbitrary ontologies.

Yawas[7] is a plugin developed for Mozilla Firefox and Google Chrome browser. The annotations are Web pages highlights, without using any semantic resource.

Table 1. Characteristics of tools. Kind of annotation (A=automatic, H=hybrid, M=manual), Saved annotation (I=intrusive, NI=non-intrusive), Platform (D=desktop, W=web)

Tool	Kind of annotation	Saved annotation	Format of input documents	Format of ontologies	Arbitrary ontology	Platform
Annotea	M	NI	Web documents	-	No	W
Annozilla	M	NI	Web documents	-	No	W
Autômeta	H	I	TXT	N-Triple, RDF, OWL, XML	Yes	D
GATE	H	NI	PDF, TXT, HTML, DOC, ODT	RDF, OWL	Yes	D
GoNTogle	H	NI	PDF, RTF, TXT, DOC, ODT	OWL	Yes	D
KIM	A	NI	HTML	RDF, OWL	No	W
Knowtator	M	NI	PDF, TXT, HTML, DOC, ODT	RDF, OWL, XML	Yes	D
Melita	M	NI	PDF, TXT, HTML, DOC, ODT	OWL	No	D
MnM	M	NI	HTML, TXT	DAML + OIL, RDF	Yes	W
Ontea	A	NI	PDF, TXT, DOC, e-mails, e-mail attachments in HTML	OWL	No	D
RDFaCE	M	I	PDF, TXT, HTML, DOC, ODT	-	No	D
RDFa Editor	A	NI	PDF, TXT, HTML, DOC, ODT	RDF, OWL, XML	Yes	D
Yawas	M	I	Web pages	-	No	W

3.2. Tools Analysis

This analysis aims to identify which tools attend the required characteristics for the semantic annotation of biomedical documents. Such characteristics include the kind of

[7] http://www.keeness.net/yawas/index.htm/

annotation and the possibility of using an arbitrary ontology for annotation. We aimed at selecting automatic annotation tools that also provide support for manual annotation. On one hand, the automation was needed due to the large volume of texts and also to the difficulty and high cost to keep specialists responsible for the manual annotation task. On the other hand, the manual annotation would be used for extra annotations, according to the user needs. Regarding the use of arbitrary ontologies, as stated before, the user needs different domain ontologies for the annotation of a biomedical text, and therefore, the tools should be flexible enough to load the selected ontologies.

In this preliminary evaluation, various tools have been dismissed for not meeting the required characteristics previously mentioned. They are: Annotea, Annozila, KIM, Knowtator, Melita, Ontea, RDFa Editor, RDFaCE and Yawas.

Despite the fact that the KIM tool was dismissed it provides a friendly interface and has an excellent support through a mailing list. However, to make annotations using an arbitrary ontology, the KIM uses GATE platform processing resources via command line. Due to that, the GATE tool was used directly.

RDFa Editor and MnM tools were discarded for technical problems. Although RDFa Editor attends the required characteristics, it is still under development and was not available for download and testing until the moment of this article closure. MnM tool, although well documented and capable of performing automatic annotation based on ontologies, presented technical limitations that precluded its use. The extraction system information necessary for the integration with the tool is no longer available on the developer portal, and it seems the project was discontinued.

GoNTogle tool presented failures in the connection with the Protégé server, this problem was not solved until the closure of this work, despite recommendations suggested by the developers. This tool could not be effectively tested, but as it has all the characteristics desired for the work, this problem has not definitively ruled out the use of this tool in future trials.

The AutôMeta and GATE tools were the only ones that did not presented failures and could participate on a further experiment, for the annotation of articles in the biomedical scenario. The experiment and its results are described in the next section.

Additionally, when conducting these tests, other problems related to the use of such tools were found. AutôMeta presented a problem using its graphical interface, however it could be used through command line. GATE failed when loading large ontologies, probably caused by problems with memory management. For the storage of non-intrusive annotation, each document must be saved individually, which can become a limitation for tests involving the annotation of a very large set of texts.

Overall, we have observed that many tools are described in order to make semantic annotation of documents, but actually they highlight the text, without reference to semantic content through ontologies or taxonomies. Another consideration relates to the lack of documentation. Many of them have only the article which describes the tool or a basic explanation on a Web page. Finally, there is also the problem of project discontinuation.

4. Experiment and Results

4.1. Scenario

AutôMeta and GATE were selected for the semantic (ontology-based) annotation experiment. As the biomedical field is very large, we restrict ourselves to the subdomain of studies on targets for drug development to combat neglected tropical diseases, which is the focus of some researchers at the Proteins and Peptides Biochemistry Laboratory at Fiocruz. To build the corpus of this experiment 108 full papers in the biomedical area were selected, extracted by searching in PubMed for keywords related to this subdomain. The files were obtained in HTML format.

PHARE ontology, which is available at NCBO BioPortal, was selected. It was chosen because it belongs to the same domain of the corpus and also for its OWL format and small size (it includes 228 classes and 83 properties, and a taxonomy depth of 5 classes), which are required to use the GATE tool. This ontology describes concepts and functions that represent the interests of pharmacogenomics relationships.

4.2. Analysis and Results

In order to analyze the output files of the executions of the AutôMeta and GATE tools, it was necessary to develop different scripts for each tool. These results were then quantitatively analyzed. The number of distinct annotated classes and terms in each article were calculated.

Figures 2 and 3 illustrate graphs comparing the number of distinct annotated terms and the number of distinct annotated classes, respectively. It is worth noting that GATE annotated more terms than AutôMeta, and that it used a larger amount of classes of the ontology. Interestingly, despite these quantitative differences, the annotation profiles are similar, with both tools featuring high peak annotation for the same documents. These high peaks indicate an adaptability of the ontology to the subdomain of the articles.

Both tools behaved similarly when the least number of terms and classes were annotated, which could be noted in articles 2;7;27;81;98 and 101. In this case, while AutôMeta annotated 33 terms and 5 classes, GATE annotated 18 terms and 31 classes. The most annotated article using AutôMeta tool was the article 77, with 165 terms. This article was the fourth most annotated in GATE (169 terms). In the case of GATE tool, article 85 was the most annotated one, with 200 terms. The same article, using AutôMeta, was the sixth most annotated (151 terms). These numbers show a slightly different behavior with respect to the most annotated articles. We also note from Figure 2, that lines get closer (almost intersect each other) when there is a low number of terms annotated in the article.

Observing the output files (annotated articles) with respect to the number classes used for annotation, for each tool, we could note that for the same part of the text, the GATE tool was capable to recognize much more classes related to the domain ontology. The average number of classes used was 27 for AutôMeta, while for GATE it was 143. The additional annotation obtained using the GATE tool was due to the fact that it is in essence a natural language processing tool, and maybe, also to its ability to use synonymy information.

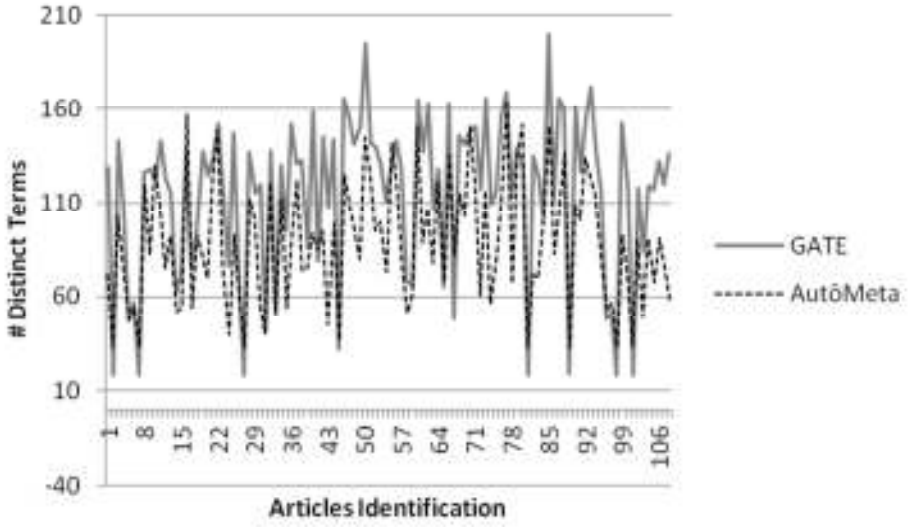


Figure 2. Comparative analysis between number of distinct terms.

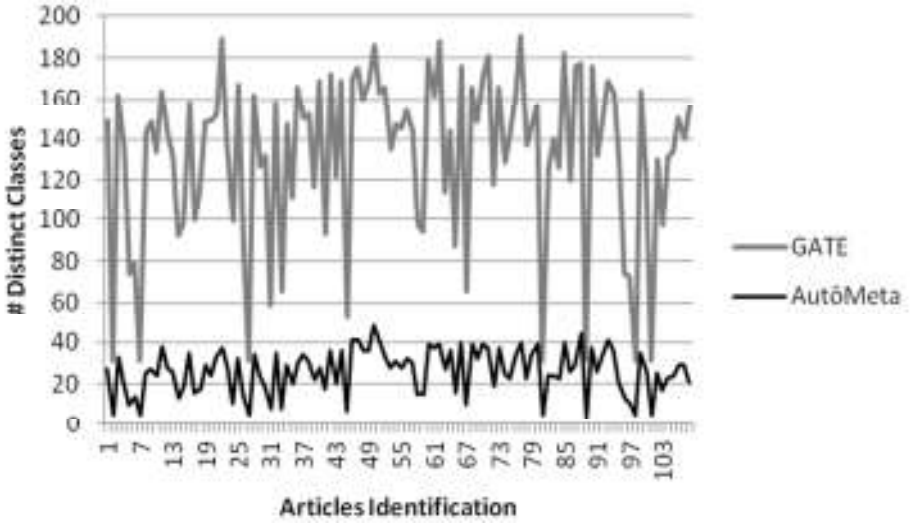


Figure 3. Comparative analysis between the number of distinct classes.

Unfortunately, it was not possible to evaluate all annotated articles in a qualitative manner. However, the article 85 was chosen to a qualitative analysis for its representativeness with respect to the number of annotated terms.

As AutôMeta annotates using RDFa standard, this facilitated the extraction of information as well-formed RDF triples (subject, predicate, object), as shown in Table 2. Moreover, Autômeta is able to make inferences, and produce additional annotations, not only for hierarchical relations, but also for domain specific relations. However, its focus is not on the use of natural language processing resources, which explains its low performance with respect to the number of annotated terms.

Table 2. Examples of RDF triples resulting from the Autômeta annotation

Subject	Predicate	Object
phare:Intramuscular	Label	Intramuscularly
phare:Organism	Label	living system
phare:Drug	Label	content
phare:Infection	Label	http://www.stanford.edu/~coulet/phare.owl#Symptom

Regarding GATE, although it does not use the RDFa standard, it generates its output in its own format, which facilitates information extraction (via script). A small part of the result extracted from the GATE annotation in article 85 can be found in Table 3. It is worth noting that GATE was able to annotate based on synonymy information (owl metadata). This is the case of "DrugSentivity", "DiseaseExacerbation" and "DrugDose" classes, which were used to annotate synonymous terms, "Tolerance", "Drug" and "G", respectively.

Table 3. Information derived from the GATE annotation

Class	Term
http://www.stanford.edu/~coulet/phare.owl#DrugSensitivity	Tolerance
http://www.stanford.edu/~coulet/phare.owl#Drug	Drug
http://www.stanford.edu/~coulet/phare.owl#DiseaseExacerbation	Growth
http://www.stanford.edu/~coulet/phare.owl#DrugDose	G

Analyzing the results of each tool annotation for article 85, we identified annotated terms that were not relevant, i.e., wrong annotations (when the used class is not related to the annotated term) and superfluous annotations (not really important to the focus of the article). Therefore, it was possible to calculate the annotation *precision*, which is the rate between the number of relevant annotations and the number of annotations (relevant/total). Both tools showed a low performance, 56% for AutôMeta and 53% for GATE. The *recall* was not calculated because it depends on what is relevant for the users, i.e., a manual annotation performed by a domain specialist would be necessary.

Although the annotations with PHARE ontology were significant, other important terms for the subdomain of the articles would also be relevant. However, PHARE did not cover all these subdomains, such as names of diseases, organisms, genes and proteins. This emphasizes, one more time, the need for annotation with multiple ontologies, so that these subdomains could also be covered. This study did not include the annotation with multiple ontologies due to problems found, as reported in the next section.

In summary, the main advantage of AutôMeta tool is that it uses RDFa standard, and that it supports the load of large ontologies such as the Molecule Role and NCI Thesaurus. On the other hand, GATE is a very solid and mature tool. Its main advantage is that it uses natural language processing resources. Both are user-friendly, but GATE is a bit more complicated at first. With respect to the inference ability, both AutôMeta and GATE include it. However, the focus of AutôMeta is on an intensive exploration of the ontology inference potential. Finally, both tools have a good documentation and are free.

4.3. Difficulties

One of the objectives of this study was to highlight the importance of semantic annotation with multiple ontologies. For this purpose, we designed an experiment where, in addition to the PHARE ontology, other ontologies were planned to be used: the Molecule Role[8] ontology and NCBITaxon taxonomy. The first refers to an ontology

[8] http://bioportal.bioontology.org/ontologies/1029/

used to annotate names of proteins and protein families, and the second refers to a taxonomic classification of living organisms. However, it was not possible to conduct such experiment because we could not load these other ontologies into the GATE tool. This was due to their large size (Molecule Role has 9,217 classes and 41.8Mb in the XML format and NCBITaxon has 513,248 classes and 243Mb in the OWL format). Therefore, we can conclude that the GATE tool is not prepared to handle large ontologies, a typical feature of biomedical ontologies. Although the AutôMeta tool presented long-term executions, it was able to make annotations with all the chosen ontologies.

A possible solution to this problem was envisaged. Dividing these ontologies into modules, taking only the parts of interest to the user, would reduce their size and facilitate their reuse. There are tools available on the web for the modularization and/or module extraction of ontologies [Garcia et al., 2012]. For such tools it is also required a large memory capacity to load them and generate the corresponding modules. Nevertheless, these tools are not prepared to deal with biomedical ontologies because they generate modules based on the names of the classes. Typically, in most biomedical ontologies, class names correspond to numeric identifiers of ontologies, and not to the corresponding terms. For example, in the Molecule Role ontology, the class name for the "enzyme" term (label) has value "IMR0000207" (enzyme identifier). Therefore, because of time restrictions, such alternative was left for future work.

5. Conclusions

This work surveys semantic annotation tools in the light of the biomedical scenario. Among the characteristics analyzed, the focus was on investigating their ability for automatic and manual annotation, their flexibility with respect to loading arbitrary ontologies, and their compliance to input/output standards.

Among a larger set of tools, AutôMeta and GATE were identified as the most adequate tools to attend the biomedical domain requirements, as both of them are able to load arbitrary ontologies and provide support for manual and automatic annotation. An experiment was then conducted to further evaluate these two tools. According to its results each tool has benefits and drawbacks. The AutôMeta tool is able to generate annotations using the RDFa standard, and to support the load of large ontologies. On the other hand, it shows a low performance with respect to the use of natural language processing resources, which is the main feature of the GATE tool. Both are user-friendly, provide inference ability, have a good documentation and are free.

From the experiment, we identified how important, in the biomedical scenario, it is to support annotation with multiple ontologies. Therefore, as future work we intend to modify the AutôMeta tool to use multiple ontologies, and new experiments will be carried out. The choice of AutôMeta is mainly due to its compliance to the RDFa standard, which facilitates the structuring of semantic data about each text and the consequent use of these data. Moreover, further improvements to AutôMeta include using additional natural language processing resources.

Acknowledgements

The authors would like to thank CNPq(309307/2009-0; 486157/2011-3), Fundação Ricardo Franco (PBIP), FAPERJ(E-26/111.147/2011; E-26/102.521/2010) for partially funding their research projects.

References

Bikakis, N., Giannopoulos, G., Dalamagas, T. and Sellis, T. (2010) "Integrating Keywords and Semantics on Document Annotation and Search". Proc. of the Int. Conf. on the move to meaningful internet systems: Part II, Hersonissos, Crete, Greece.

Ciravegna, F., Dingli, A., Petrelli, D. and Wilks, Y. (2002) "Timely and Non-Intrusive Active Document Annotation Via Adaptive Information Extraction". In Semantic Authoring, Annotation and Knowledge Markup (SAAKM02), ECAI.

Cunningham, H., Maynard, D., Bontcheva, K. and Tablan, V. (2002) "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". Proceedings of the ACL.

Ding, Y., Embley, D.W. and Liddle, S.W. (2006) "Automatic Creation and Simplified Querying of Semantic Web Content: An Approach Based on Information-Extraction Ontologies". Proceedings of the First Asian Semantic Web Conference (ASWC'06). Berlin Heidelberg: Springer, pp. 400-414.

Duma, M. (2011) "RDFa Editor for Ontological Annotation". Proceedings of the Student Research Workshop associated with RANLP 2011, pages 54–59, Hissar, Bulgaria.

Fontes, C.A. (2011) "Explorando Inferência em um Sistema de Anotação Semântica", Master's thesis, Dept. of Computer Science, Military Inst. of Engineering, Rio de Janeiro, Brazil.

Garcia, A.C., Tiveron, L., Justel, C., Cavalcanti, M.C. (2012) "Applying Partitioning Algorithms to Modularize Large Ontologies". In: Proc. of ONTOBRAS.

Kahan, J., Koivunen, M., Prud'Hommeaux, E. and Swick, R.R. (2001) "Annotea: An Open RDF Infrastructure for Shared Web Annotations", Proceedings. of the WWW10 International Conference, Hong Kong.

Khalili, A. and Auer, S. (2011) "The RDFa Content Editor - From WYSIWYG to WYSIWYM". http://svn.aksw.org/papers/2011/ISWC_RDFaEditor/public.pdf, September.

Laclavik, M., Seleng, M., Gatial, E., Balogh, Z. and Hluchy L. (2006) "Ontology based Text Annotation – OnTeA". Proceedings of 16-th European-Japanese Conference on Information Modelling and Knowledge Bases (EJC'2006), pp. 280-284, Trojanovice, Czech Republic.

Noy, N. F., Shah, N.H., Whetzel, P.L. et al. (2009) "BioPortal: Ontologies and Integrated Data Resources at the Click of a Mouse". Nucleic Acids Res. Jul 1;37(Web Server issue):W170-3. PMID: 19483092.

Ogren, P.V. (2006) “Knowtator: a Plug-in for Creating Training and Evaluation Data Sets for Biomedical Natural Language Systems”. Proceedings of the 9th International Protégé Conference 73–76.

Popov, B., Kiryakov, A., Ognyanoff, D., et al. (2003a) “Towards Semantic Web Information Extraction”. Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003), Florida, USA, 20 October.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. and Goranov, M. (2003b) “KIM - Semantic Annotation Platform”. In 2nd International Semantic Web Conference, (Florida, USA, 2003), 834-849.

Shadbolt, N., Hall, W. and Berners-Lee, T. (2006) “The Semantic Web Revisited”. IEEE Intell. Syst., 21(3): 96-101.

Smith, B., Ashburner, M., Rosse, C., et al. (2007) “The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration”. Nature Biotechnology 25, 1251 - 1255.

The Gene Ontology Consortium. (2000) “Gene ontology: Tool for the Unification of Biology”. Nat. Genet., 25(1):25-9.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. and Ciravegna, F. (2002) “MnM: Ontology Driven Semi-automatic and Automatic Support for Semantic Markup”. Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, p.379-391.

A Tool for Efficient Development of Ontology-based Applications

Olavo Holanda[1], Ig Ibert Bittencourt[1], Seiji Isotani[2], Endhe Elias[1], Judson Bandeira[1]

[1]Computing Institute – Federal University Alagoas (UFAL)
Maceió – AL – Brazil

[2]Department of Computer Science, Institute of Mathematics and Computer Science
University of São Paulo, São Carlos, São Paulo, Brazil

{olavo.holanda,ig.ibert, endhe.elias, jmb}@ic.ufal.br, sisotani@icmc.usp.br

Abstract. *In the past few years, the use of ontologies for creating more intelligent and effective application has increased considerably. This growth is due to the fact that ontologies attempt to provide semantics to the data consumed by machines, so that they can reason about these data. However, the development of applications based on ontologies is still difficult and time-consuming, because the existing tools lack to provide a simple and unified environment for the developers. Most of these tools only provide data manipulation using RDF triples, complicating the development of applications that need to work with the object orientation paradigm. Furthermore, tools which provide instances manipulation via object orientation do not support features such as manipulating ontologies, reasoning over rules or querying data with SPARQL. In this context, this work proposes a tool for supporting the efficient development of ontologies-based applications through the integration of existing technologies and techniques. In order to evaluate the benefits of this tool, a controlled experiment with eight developers (unfamiliar with ontologies) was performed comparing the proposed tool with another one used by the community.*

1. Introduction

Recently, ontologies had obtained quite attention in the computer scientific community. The term, which has origins in philosophy, becomes a useful word in computer science for a new approach of knowledge representation about real-world entities. For this, ontologies offer a shared understanding of a particular domain and a formalization which allows its data to be interpretable by machines[Hepp et al. 2007].

As a result, ontologies are not only applied as basis for the so called Semantic Web, but in other areas of computing research and industry. For example, e-commerce applications use ontologies for parametric searches and heterogeneous systems integration[Das et al. 2002]. Another industry segment is the media systems that has used this approach to do real-time data inference, delivering up-to-date content for its users[Kiryakov et al. 2010]. In addition, several other fields are using ontologies such as medicine [Bard and Rhee 2004], computer mobile [Cheyer and Gruber 2010] and adaptative education [Bittencourt et al. 2009][Bittencourt et al. 2006][Bittencourt and Costa 2011].

This dissemination is a consequence of the growing number of tools and software libraries that allow the development of semantic applications. Currently, more than 170

tools are listed at the semanticweb.org, and this number tends to increase. Despite the high number of tools, not all of them aim to support the development of applications for the Semantic Web. On the other hand, the tools that offer this type of support do not provide it through a simple and unified environment, that is, they fail to offer common functionalities when developing applications based on ontologies, such as managing and querying ontologies, reasoning over rules, manipulating instances via object oriented paradigm (in contrast to the manipulation of instances via triple RDF - *Resource Description Framework*), among others.

In this context, this paper proposes a tool that provides simplified development of ontologies through the object oriented model. Moreover, the tool provides an integration of existing technologies and techniques to create a unified environment for developers of applications based on ontologies. This tool provides services such as operations on ontologies, manipulating instances, SPARQL[1] (*Simple Protocol and RDF Query Language*) queries, data inference over SWRL[2] (*Semantic Web Rule Language*) rules, and so on. This paper is organized as follows. In Section 2, the characteristics of each type of ontology programming are outlined. In Section 3, the proposed tool is described. The performed experiment, evaluating the proposed work is presented in Section 4. Section 5 presents some conclusions and future works.

2. Ontology Programming

During the development of a semantic-based application, the manipulation of instances is one of the steps in the process of development. For this step, there are currently two main approaches used by the ontology management systems: RDF triples and object oriented development. In the next subsections, the main distinctions and benefits of the aforementioned approaches are detailed.

2.1. RDF Triples Development

Most current APIs (*Application Programming Interface*) are still working with the development based on RDF triples (subject, predicate and object). Thus, application developers should be aware of how the ontology works in RDF layer, in order to manipulate the data through each triple in application code.

When the developer desires to add a resource (subject) in the ontology with several properties inherent to it, several lines of code representing each triple of the resource will be necessary. Each triple captures a single value of each property. Similarly, if the application removes this resource, several triples should be removed.

For example, to create an instance *alice* of the entity Person with the datatype property *name* “Alice” using the Sesame API, which works with the development based on RDF triples, several lines of code are needed. Figure 1 shows how to add this resource in the Sesame repository.

[1]SPARQL Query Language for RDF is W3C recommendation since January 2008 - `http://www.w3.org/TR/rdf-sparql-query/`

[2]More information at: `http://www.w3.org/Submission/SWRL/`

```
...
ValueFactory f = myRepository.getValueFactory();

// create some resources and literals to make statements out of
URI alice = f.createURI("http://example.org/people/alice");
URI name = f.createURI("http://example.org/ontology/name");
URI person = f.createURI("http://example.org/ontology/Person");
Literal alicesName = f.createLiteral("Alice");

RepositoryConnection con = myRepository.getConnection();
// alice is a person
con.add(alice, RDF.TYPE, person);
// alice's name is "Alice"
con.add(alice, name, alicesName);
...
```

Figure 1. Code example from Sesame API

2.2. Object Oriented Development

Instead of RDF triples, object-oriented applications manipulate data at object level and their attributes. Such objects are characterized by a set of attributes and values. In this sense, a tool is necessary to "mapp" the operations on objects to the RDF triples infrastructure that works underneath. Some tools were created to provide such paradigm for handling instances in ontologies.

As a result, developers do not need to have a deep knowledge of the ontology representation language. An object in the code represents an instance in the ontology, their attributes are mapped to the properties of the instances and RDF classes become Classes in the programming language. Therefore, to add a resource in the ontology, the developer just need to add the object, facilitating, in this way, the development of such applications.

Comparing to Sesame example, Figure 2 shows the same resource added in the Sesame repository, but now using Alibaba API, which allows the development based on the object-oriented paradigm.

```
...
ObjectConnection con = repository.getConnection();

// create a Person
Person alice = new Person();
alice.setName("Alice");

// add a Person to the repository
con.addObject(alice);
...
```

Figure 2. Code example from Alibaba API

3. Proposed Tool

Currently, there are several tools which manipulate ontology through the paradigm of object orientation (e.g. Jastor [Szekely and Betz 2009] and Elmo [Mika 2007]) instead of RDF triples. However, these tools provide only the manipulation of instances, which is only one step of the ontology-based development process. Therefore, when working with these tools, developers need to search other libraries to build semantic applications with usual features (e.g. query ontologies, handling instances, perform rules, etc.). To alleviate this lack of appropriate development tools to build and maintain semantic applications, this paper proposes a tool (more specifically a Java software library) that integrates several features that facilitate the development based on ontologies.

The system requirements of the proposed tool are:

- **Repositories Handling**: The first service needed for ontology-based development that the tool provides is the remote management of Sesame repositories. This management is composed of creation and delete of repositories in a Sesame server;
- **Persistence of ontologies**: The tool also provides, for the user, ontologies persistence service in repositories. OWL or RDF files can be added to a repository, which stores all data in binary files. Besides adding ontologies, this service allows the delete and retrieve of a given ontology from repositories;
- **Handling Instances**: The main service provided by the tool is the manipulation of ontology instances present in a repository. Note that this manipulation is done through the paradigm of object orientation. This service is composed of the methods for creating, retrieving and removing instances of a repository;
- **Generation of Java code**: To improve the instances manipulation following the paradigm of object orientation, the automatic generation of Java code from ontologies is required. This feature allows developers to "map" ontologies in Java code, creating classes that represent entities and concepts in these ontologies. As a final result of this feature, a Java library is created,
- **SPARQL queries**: When developing any application, it is very common to query the data stored by it. It is not different when the application is based on ontologies, where developers need a service that can query RDF graphs. The proposed tool provides a service which offers queries on ontologies based on SPARQL language;
- **Backup Repository**: Developers often need, for some industry applications, to keep backups of a repository. This feature allows the creation of backup files, recovering these files into an empty repository and the copy between different repositories;
- **Verification and Validation of Ontology**: A service that allows consistency checking of an ontology. In other words, this feature verifies that an ontology does not contain contradictions, i.e., if there is only one interpretation of the concepts in the ontology;
- **Reasoning over Rules**: This service allows the performing of SWRL rules present in the repository, creating new information through the reasoning over them.

3.1. System Architecture

The system architecture is based on the layers pattern [Buschmann et al. 2007], Figure 3, where each layer uses only the services of the layer below. The architecture will be detailed using a bottom-up approach, starting with the layer OWLIM. OWLIM is a set

of semantic repositories of high performance and scalability[Kiryakov et al. 2005]. The proposed tool offers the repository creation with default configuration as OWLIM *Lite* version. The OWLIM was used along with the Sesame server, so that these repositories can be accessed remotely. The integration between Sesame and OWLIM is done by the OWLIM tool, which implements an extension of the Sesame Repository API.

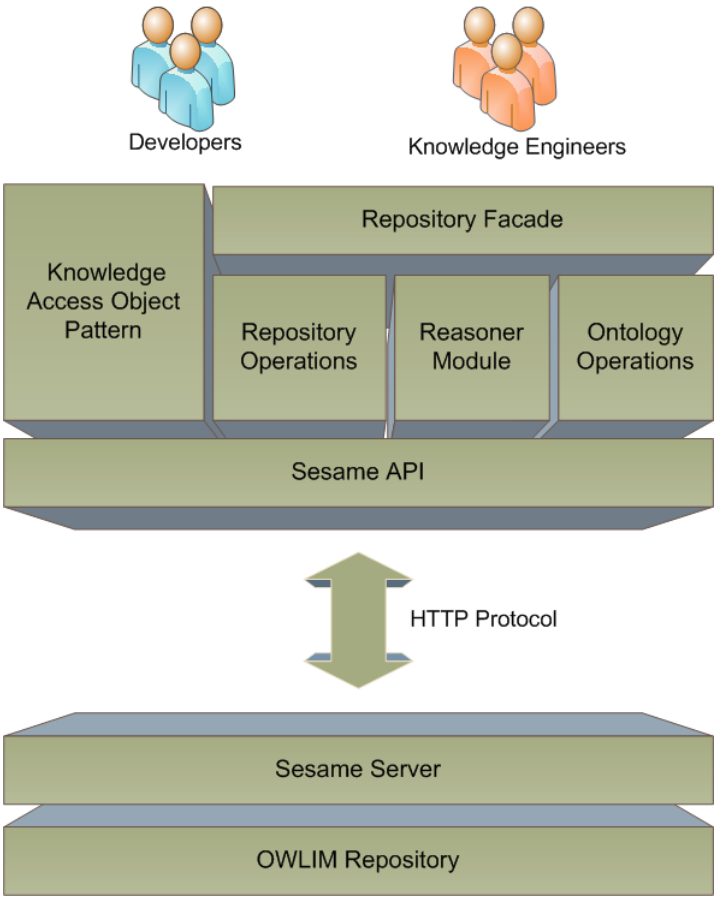


Figure 3. System Architecture

To access a repository in Sesame server, it is necessary to use the Sesame Repository API, which behaves in this case as a client in a client-server architecture [Buschmann et al. 2007]. This API communicates via HTTP (*Hypertext Transfer Protocol*) with Sesame server. In this layer are the methods required to connect to the repository and data manipulation through RDF triples. Above this layer, there are four modules: operations on ontologies, reasoning module, operations on repositories, and the KAO pattern. The layer "operations in ontologies" is composed of functionalities performed on ontologies (ontology itself, not the instances). Among these features are: the insertion and delete of an ontology, generate Java code from one or more OWL files and ontology validation.

On the other hand the layer of "operations in repositories" brings together relevant services for manipulating Sesame repositories. This module allows the developer to operate in remote repositories, such operations can be: the creation or removal of a repository, clear a repository (delete all data present on it) and create backup copies of a given repository.

The next layer to be detailed is the "reasoning module". This module has the goal to infer new data in a repository by executing SWRL rules present on it. Once you run this module, it will look if there are rules in the repository, if any, it will check for new data to be inferred from the execution of those discovered rules. See on Figure 1 that these last three discussed layers are under the "repository facade" layer, which is designed to provide a unified access to these three modules through the Facade design pattern [Gamma et al. 2004].

Last but not least, comes the layer called "Knowledge Access Object" (KAO), which is a persistent pattern similar to the Data Access Object (DAO), with the difference that the KAO not only works with data, but works with information from the ontologies. The KAO pattern aims to provide an abstraction of the persistence mechanism used, providing some specific operations (such as creation, retrieval and removal of instances, among others) without exposing details about connections to the repository. Thus, this pattern can isolate the persistence layer of the business layer. In the next section, this pattern will be illustrated.

3.2. KAO Pattern

The module has an abstract class called *AbstractKAO* responsible for all the abstraction of the KAO pattern. It provides operations for manipulating instances (create, remove, and retrieve) and performing queries. The query methods of the abstract class are protected, that is, only a class that extends AbstractKAO can use these methods. This is intentional, so that the KAO pattern works properly.

Exemplifying the KAO pattern from the user's perspective (Figure 4), who wants to manipulate instances in a given ontology A. This user must create a concrete class (called *OntologyAKAO*) that extends *AsbtractKAO*. This class has concrete methods of creation, retrieval and removal of instances inherits from the abstract class. If the user wants to make a query, he performs the query inside the class *OntologyAKAO*, isolating thereby the persistence layer of the other application layers.

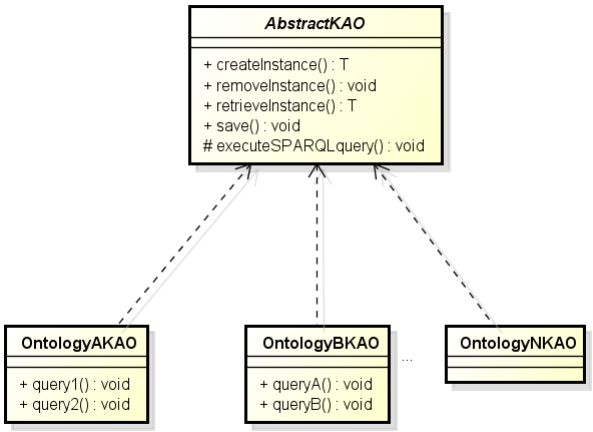


Figure 4. KAO Persistence Pattern

4. Experiment

This section proposes an experiment which analyzes the proposed tool, provides a quantitative and qualitative evaluation. Moreover, the experiment does a comparison between our tool and a tool available in the literature. This comparison aims to verify if the obtained results satisfy the initial proposal, in order words, if the proposed tool increases the efficiency of the programmer development. The experiment was based on [Wohlin et al. 2000] and it has four steps: definition, planning, running and analysis.

4.1. Definition

The first step is determinate the problem which will be analyzed. For this work, the goal is to evaluate the differences on the programmer development. This work does a comparative analysis between the cost to implement a semantic application using the proposed tool and the cost to implement the same application using another tool named Jastor. The experiment occurred at university context and it was done in January 2012. The experiment had eight participants and it focused on the comparison between the proposed tool and the Jastor tool.

4.2. Planning

After the definition of scenario, the next step is planning. Thus, the planning was divided into three main activities: i) creation of control groups; ii) specification of application which will be developed; and iii) specification of the evaluation metrics. The experiment also evaluates the proposed tool and the Jastor tool together with Jena. The choice of these tools is due to two reasons. The first one is that Jena is one of the most popular APIs for manipulating ontologies in Java. The second reason is the tool Jastor is the main tool which works with ontologies in object level with Jena. Therefore, there was this need to integrate the two tools (Jastor and Jena) for this experiment.

The first activity of planning step is the creation of control groups. This experiment had eight undergraduate students of Computer Science and Computer Engineering courses. Several lessons were taught in order to equate the knowledge of participants, thus increasing the experiment control. The lessons were about: i) object-oriented programming and Java language; ii) ontologies modeling using the Protégé environment; iii) queries on ontologies using SPARQL language; and iv) inference on ontologies using SWRL. None of the participants built any ontology-based application before the experiment.

The experiment environment was divided into four control groups, where each group had a pair of students. The first group was named **F5** and its pair used machines with Intel Core i5 CPU processor and 4GB of RAM. The second group was named **F3** and its pair used machines with an Intel Core i3 CPU processor and 4GB of RAM. The **F5** and **F3** groups used the proposed tool. On the other hand, **J5** and **J3** groups used the tools Jastor/Jena and their machines were the same of **F5** and **F3**, respectively. All machines in the experiment performed the operating system Windows 7 Professional 64-bits. The Table 1 summarizes the experiment environment.

After the groups creation, the second activity is the specification of the application which will be developed by the participants. During the experiment, each group will develop the same ontology-based application using its allocated tool. This application

Group	Machine	Allocated Tool
F5	Intel Core i5 CPU processor and 4GB of RAM	Proposed Tool
F3	Intel Core i3 CPU processor and 4GB of RAM	Proposed Tool
J5	Intel Core i5 CPU processor and 4GB of RAM	Jastor/Jena Tool
J3	Intel Core i3 CPU processor and 4GB of RAM	Jastor/Jena Tool

Table 1. The Experiment Environment

uses an adaptation of the *Family.swrl.owl*[3] ontology. This ontology is included in Protégé ontology libraries and it aims to demonstrate the use of SWRL rules in ontology about family relationships. The same Java project will be given to all groups and it contains the main method (*Main*) with features not implemented. The groups goal is run the *Main* method. Therefore, the groups must implement the necessary infrastructure using the allocated tool. The *Main* method has common steps in the development of ontology-based applications, such as: i) add the ontology in repository/database; ii)manipulate instances of ontology; iii)run the SWRL rules; and iv) do SPARQL queries.

The last activity of the planning of the experiment is the specification of the evaluation metrics. As aforementioned, the experiment focuses on the developing efficiency of the participants. Therefore, some quantitative metrics were defined, such as: i) development time, measured in hours; ii) number of lines of implemented code, in this case were not counted: the *Main* lines, the commented lines and the blank lines; iii) performance, measured in milliseconds, iv) memory usage, measured in *Megabytes*; and v) number of errors encountered during development, these errors are Java exceptions that were thrown when a method of the tools was run incorrectly.

In order to have a subjective evaluation about the proposed tool, a questionnaire was elaborated which has questions about the experience of each developer after the end of the experiment. The quantitative and subjective results will be used in the comparison between the tools. The questions are:

1. What did you think about the tool documentation?
2. What did you think about the tool setup?
3. What was the complexity level in the addition of ontology on repository?
4. What was the complexity level in the manipulation of instances?
5. What was the complexity level in the running of the SWRL rules?
6. What was the complexity level in the running SPARQL queries?
7. What did you think that overall the tool?

4.3. Running

The experiment started on January 23th, 2012 and it finished on January 26th, 2012. On the first morning, the experiment was introduced to the participants. At this moment, a general explanation was presented about the experiment and this moment was the first contact between experiment and participants. After the explanation, the pairs were formed and the tools were allocated to each pair (see Table 4.2). Furthermore, the links[4] were

[3]Available at: `http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl`

[4]Jastor and Jena documentation: http://jastor.sourceforge.net/ and http://jena.sourceforge.net/documentation.html. Proposed tool documentation: http://jointnees.sourceforge.net/tutorial.html

provided which present the documentation of the tools.

The first data were collected during the experiment. Each pair reported the development time and number of errors encountered. When some team can run the Main method without errors, their code is evaluated. After evaluation, the number of implemented code lines was collected. This count was done manually. On the other hand, the JConsole[5] tool was used to collect the data related to performance and memory usage.

4.4. Analysis of Results

For providing more valuable information about the two target tools, the Table 2 presents the data related to quantitative metrics.

Group	Development Time	Code lines	Performance	Memory Usage	Number of Erros
F5	7 h	72 lines	2584 ms	15,4 MB	3 errors
F3	6 h	81 lines	3757 ms	16,2 MB	1 error
J5	15 h	89 lines	4070 ms	61,5 MB	11 errors
J3	18 h	84 lines	4144 ms	52,2 MB	5 errors

Table 2. The Quantitative Data of the Experiment

Regarding the development time, the proposed tool showed a lower time than the tool Jastor/Jena. On the one hand, the F3 pair finished the experiment in 6 hours and the F5 pair in 7 hours. On the other hand, the J3 pair finished in 18 hours and J5 in 15 hours. The development time is one of the most important factors to evaluate the efficiency on development of ontology-based application. Thus, the proposed tool helps the developers in this issue.

The third column (Table 2) presents the number of code lines for each team. Although, the difference between the pairs was very close when writing code, the proposed tool presented to be considerably more optimized in this issue. This is due to the high level of abstraction which the proposed tool provides.

Then, two factors (performance and memory) are analyzed. Although, these factors are not directly related to efficiency in development, they are very important in the performance of the built application. The performance and the memory usage are measured when the *Main* is running. The performance of the proposed tool was better when we comparing the teams F3 and J5. In other words, the proposed tool had shown the higher performance using a lower machine. The best performance was the F5 team (2.5 seconds) and the worst was J3 team (4.1 seconds). Related to the memory usage, the proposed tool was better because the pairs who used it obtained a significantly smaller cost than the other two pairs. The number of errors metric can be considered the best issue of this work, because the groups who used the proposed tool found four times less number of errors than the others, who used Jastor/Jena.

The Table 3 presents the results of qualitative evaluation. The participants gave a standard note on each question where 1 is the lowest and 5 is the highest. Depending on the question, 1 means terrible and 5 means great. The results show that the proposed tool surpasses the Jastor/Jena tools and they are a reflection of the quantitative results.

[5]Available at: `http://openjdk.java.net/tools/svc/jconsole/`

	Tool	Note 1	Note 2	Note 3	Note 4	Note 5
Question 1	Proposed Tool	0	0	0	4	0
	Jastor/Jena	0	1	2	1	0
Question 2	Proposed Tool	0	0	0	2	2
	Jastor/Jena	0	1	2	1	0
Question 3	Proposed Tool	0	0	0	0	4
	Jastor/Jena	1	2	1	0	0
Question 4	Proposed Tool	0	0	0	3	1
	Jastor/Jena	0	0	3	1	0
Question 5	Proposed Tool	0	0	4	0	0
	Jastor/Jena	1	2	0	1	0
Question 6	Proposed Tool	0	0	0	2	2
	Jastor/Jena	0	0	0	2	2
Question 7	Proposed Tool	0	0	0	2	2
	Jastor/Jena	0	1	1	2	0

Table 3. The Quantitative Data of the Experiment

5. Conclusions and Future Works

This paper aimed to propose a tool to unify several features (manipulate instances, operations on ontologies and repositories, SPARQL query support and inference over SWRL rules) necessary for the development of applications based on ontologies. The tool was presented through its architecture and services description.

As a result, an experiment was conducted to evaluate the tool, focusing on the efficiency of the programmer development. The experiment compared the proposed tool with the tools Jastor and Jena, and both the objective analysis (development time, lines of code, performance, memory and number of errors) and the subjective analysis (through forms) showed better results for the proposed work. Furthermore, although not yet presented in the literature, versions of the proposed tool were used in some works already published, such as [da Silva et al. 2011], [Ferreira et al. 2011], [Holanda et al. 2012] and [Ataide et al. 2011].

However, there are some issues in this work that still need to be addressed. The conducted experiment should be extended to other tools, such OWL2Java and Protege-OWLAPI and with a greater number of developers. The SWRL algorithm must be improved aiming a better performance and further features can be added to the presented tool.

References

Ataide, W., Brito, P., Pedro, A., Costa, E., and Bittencourt, I. I. (2011). A semantic tool to assist authors in the instantiation of software product lines for intelligent tutoring systems context. In *Proceedings of the 4th Workshop on Semantic Web and Education*. WSWEd.

Bard, J. B. L. and Rhee, S. Y. (2004). Ontologies in biology: design, applications and future challenges. *Nature Reviews Genetics*, 5(3):213–222.

Bittencourt, I., Bezerra, C., Nunes, C., Costa, E., Tadeu, M., Nunes, R., Costa, M., and Silva, A. (2006). Ontologia para construção de ambientes interativos de aprendizagem. *Anais do Simpósio Brasileiro de Informática na Educação*, 1(1):567–576.

Bittencourt, I. and Costa, E. (2011). Modelos e ferramentas para a construção de sistemas educacionais adaptativos e semânticos. *Revista Brasileira de Informática na Educação*, 19(01):85.

Bittencourt, I. I., Costa, E., Silva, M., and Soares, E. (2009). A computational model for developing semantic web-based educational systems. *Knowledge-Based Systems*, 22(4):302 – 315.

Buschmann, F., Henney, K., and Schmidt, D. (2007). *Pattern-oriented software architecture: On patterns and pattern languages*. Wiley series in software design patterns. Wiley.

Cheyer, A. and Gruber, T. (2010). Siri: A virtual personal assistant for iphone, an ontology-driven application for the masses. Presentation at Open, International, Virtual Community of Practice on Ontology, Ontological Engineering and Semantic Technology.

da Silva, A., Bittencourt, I., Ataide, W., Holanda, O., Costa, E., Tenório, T., and Brito, P. (2011). An ontology-based model for driving the building of software product lines in an its context. In Dicheva, D., Markov, Z., and Stefanova, E., editors, *Third International Conference on Software, Services and Semantic Technologies S3T 2011*, volume 101 of *Advances in Intelligent and Soft Computing*, pages 155–159. Springer Berlin / Heidelberg.

Das, A., Wu, W., and McGuinness, D. L. (2002). *The emerging semantic web: selected papers from the first Semantic Web Working Symposium*, volume 75 of *Frontiers in artificial intelligence and applications*, chapter Industrial Strength Ontology Management, pages 101 – 118. IOS press.

Ferreira, R., Holanda, O., Melo, J., Bittencourt, I. I., Freitas, F., and Costa, E. (2011). An agent-based semantic web blog crawler. In *Proceedings of the 7th International Conference on Information Technology and Applications*. ICITA, Sydney.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (2004). *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.

Hepp, M., Leenheer, P., Moor, A., and Sure, Y. (2007). *Ontology management: semantic web, semantic web services, and business applications*. Semantic web and beyond. Springer.

Holanda, O., Ferreira, R., Costa, E., Bittencourt, I. I., Melo, J., Peixoto, M., and Tiengo, W. (2012). Educational resources recommendation system based on agents and semantic web for helping students in a virtual learning environment (to appear). *International Journal of Web Based Communities*.

Kiryakov, A., Bishop, B., Ognyanoff, D., Peikov, I., Tashev, Z., and Velkov, R. (2010). The features of bigowlim that enabled the bbc's world cup website. In Krummenacher, R., Aberer, K., and Kiryakov, A., editors, *In proceendings Workshop of Semantic Data Management (SemData)*.

Kiryakov, A., Ognyanov, D., and Manov, D. (2005). Owlim - a pragmatic semantic repository for owl. In Dean, M., Guo, Y., Jun, W., Kaschek, R., Krishnaswamy, S., Pan, Z., and Sheng, Q., editors, *Web Information Systems Engineering - WISE 2005 Workshops*,

volume 3807 of *Lecture Notes in Computer Science*, pages 182–192. Springer Berlin / Heidelberg.

Mika, P. (2007). *Social networks and the Semantic Web*. Semantic web and beyond. Springer.

Szekely, B. and Betz, J. (2009). Jastor: Typesafe, ontology driven rdf access from java. Available from http://jastor.sourceforge.net.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA.

Sentiment analysis in social networks: a study on vehicles

Renata Maria Abrantes Baracho,
Gabriel Caires Silva, Luiz Gustavo Fonseca Ferreira

[1]Progama de Pós-Graduação em Ciência da Informação (PPGCI)
Universidade Federal de Minas Gerais (UFMG)
P.O. 486 31270-901 – Belo Horizonte – MG – Brazil

Abstract. *This paper presents partial results of a research project that aims to create a process of sentiment analysis based on ontologies in the automobile domain and then to develop a prototype. The process aims at making a social media analysis, identifying feelings and opinions about brands and vehicle parts. The method that guided the development process involves the construction of ontologies and a dictionary of terms that reflect the structure of the vocabulary domain. The proposed process is capable of generating information that answers questions such as:* "In the opinion of the customer, which car is better: Corsa or Palio? Which one is more beautiful? Which engine is stronger?" *To answer these questions by comparison, one can show a general view reflected on different social networks, indicating, for example, that for a given vehicle, a certain percentage of responses are considered positive, while for others, the percentage is considered negative.*
The results can be used for various purposes such as guiding decisions to improve the products or directing specific marketing strategies. The process can be generalized and applied to other areas in which organizations are interested in monitoring views expressed about their products and services.[1]

1. Introduction

The increase of personal information available on the Web, especially in recent years, is noteworthy at least. With the advent of what is called *Web 2.0*, countless opinions and feelings about every subject, are wildly available throughout the Web. In this new era, besides the content offered by companies and organizations, individuals have come to share reviews and opinions via personal blogs, networking sites, and microblogs, just to name a few.

This paper presents the initial results of a research project whose main objective is to create a model of knowledge representation in the context of social networks on the Web. In this project we developed a prototype software for sentiment analysis of an automobile brand on the Web. This is archived by the use of morphologic analysis, and language features detection aided by ontologies. Specific objectives include the design of methodologies for opinion mining, composition and classification, creation of a dictionary of terms that contains sentiment orientation by translating this type of dictionary from another language, design and use of ontologies to be used in the process of sentiment detection and data summarization and finally the working prototype itself.

[1]This work is partially supported by Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Governo do Estado de Minas Gerais, Brazil, Rua Raul Pompéia, nº101 - São Pedro, Belo Horizonte, MG, 30.330-080, Brazil.

Our prototype is applied to a specific company in the automobile market (FIAT) and presents innovative nature of monitoring business intelligence and user opinion. It targets information available on the Web from several sources (such as automotive centered portals, blogs and discussion groups) in Portuguese language, although the presented methodology could be easily applied to any other language, source or targeted domain. The objective for the prototype is that it should be able to answer questions and give important insights about the sentiment on car brands on the Web. For example: *"what are people saying about FIAT Punto in social media?"* These results can be used to improve products or direct marketing strategies, as well as be applied by any other organization interested in monitoring sentiment about a product and/or service.

The proposed methodology and our resulting prototype collects, structures and analyzes Web information by using a combination of text processing technologies with several other linguistic techniques such as morphological, syntactic and semantic analysis guided by the target domain terms supplied by our ontologies and the list of terms that evokes sentiment (with a given polarity and strength value) supplied by our sentiment dictionary. The information in our ontology is structured as trees of objects that relate to each other as a *is part of* or *is one* relationships. Each object may have one or more terms that can be used to identify references to that particular object on the sentences extracted.

Each sentiment detected in the process described is then stored in the ontology tree structure, as a link of one or more objects in the ontology tree to a sentiment value, that can be either neutral, positive or negative. Using this hybrid unsupervised approach, by combining language processing, lexicon techniques and ontology techniques for the sentiment data structure, we expect to generate classifiers for sentiments and opinions and business intelligence insights that improve the results obtained so far in sentiment analysis and opinion mining without relying on supervised algorithms such as machine-learning approaches that requires a costly training phase that may be impeditive for groups with limited resources.

This project is funded by the state financing agency. The group is composed of two PhD and two graduate researchers. The project's main area of research is Information and Knowledge Management, but as our main objective is to retrieve information from texts in natural language processing them lexically and morphologically to extract semantic proprieties (sentiments) with the use of ontologies we also relate the areas of Information Extraction and Retrieval, Knowledge Representation, Analysis of Information Systems, Semantic Technology and Philosophy of Information.

2. Theoretical Framework

With the rise of Web 2.0 and specialized portals, blogs, and social networks, an enormous amount of new personal opinion is made accessible on a daily basis. Reviews, ratings, recommendations and other forms of expression are available on-line. Information previously obtained through a costly and time consuming process of satisfaction and opinion research can now be obtained on a large scale on the Web. The new challenge is how to process and interpret this massive amount of information, and this challenge is the object of research in the discipline called "sentiment analysis and opinion mining". In this research we consider the following definition for the term "sentiment analysis and opinion mining:" the identification, extraction and study of opinions, feelings and emo-

tions expressed in texts from the web. In the following section, the theoretical framework of the research project can be found. The research papers reviewed below inspired the developments of the method and process of sentiment analysis detailed in Section 3.

2.1. Ontologies

[Cicortas et al. 2009] There is a growing demand for information systems-oriented interpretation of human language. These systems are designed to be capable of understanding the intentions and opinions of the author with minimal human intervention. In the article entitled Considerations on Ontologies Construction, the author identifies the challenge the interpretation of heterogeneous information by automated tools and analyzes possibilities of using ontology to resolve these issues. The combination of ontological and natural language rules are seen as a solution to improve performance of sentiment analysis.

Also in this context, the importance of ontologies in identifying the meaning of information, through detailed description of complex systems is highlighted, [Rösner and Kunze 2003]. Also discussions about the best practices for building ontologies. The authors present their experiences related to construction of new ontologies, detailing various methods for the use of language constraints, design principles and ideas for frameworks. They emphasize the importance of having a quality system to detect synonyms in a process of creating ontologies, since its absence in many cases, can compromise the quality of the results.

[Polpinij and Ghose 2008] In an article *An Ontology-Based Sentiment Classification Methodology for Online Consumer Reviews*, classification is presented as a proposed ontological approach based on lexical variation. The authors propose the use of three sources for the construction of an ontology: a dictionary, a list of text and a set of verbs. From these sources the ontology is built based on three types of information: morphological analysis (indicating a pattern in the composition of the word), a parse (containing information about their classification, e.g. verbs and suffixes such as e.g. and e.s.), and finally a semantic analysis based on logical constraints of synonymy, antonymy and subsumption (relationship "is one").

The ontological structure derived is then used to create a model BOW ("bag of words") and fed into a classifier. According to the authors, this technique achieved satisfactory results, reaching 96% accuracy. [Kunze and Rösner 2005] present a methodology for ontology extension using concepts derived from a specific domain. The method uses a first and a body ontology partially processed in the domain. The approach is based on syntactic and grammatical structures and basically explores features of the language contained in the input corpus.

2.2. Sentiment Analysis

[Liu 2010] presents an introduction to key problems and solutions within the existing area of sentiment analysis research highlighting its importance both to individuals and to companies in market research and interest / customer satisfaction. The text provides important definitions such as the concepts of object and features (properties or parts of an object) and opinion (feeling positive, negative or neutral in relation to an object or a feature).

[Wang et al. 2011] propose a method of selection of features for the classification of feelings. Based on linear discriminant analysis (Fisher's discriminant ratio), the method utilizes the concept of information gain (Information Gain) and is validated through the comparison with other methods based on this concept. In the article, the authors present the results of two experiments in which selection methods tested different features. The experimental results indicated that the linear discriminant method has better performance than the others analyzed.

The approaches used by [Ramanathan and Ramnath 2010] explore the use of context in sentiment analysis using three techniques. The first is an approach that makes use of domain ontology mapping sentences on objects in the ontology. For each object, a weight is defined positive and negative and the positive and negative score of a sentence is defined as the sum of these weights. The weights are defined using machine learning techniques and regression. The second approach makes use of a technique for capturing sequences of characters that appear frequently. For each pair of sentences, extracted a set of words that appear in both, and each of these sets, you assign a score positively or negatively according to how often they appear in sentences. Finally, two approaches are used to combine three techniques for the classification of polarity of a sentence and the results presented.

[Wei and Gulla 2010] present analysis technique based on a tree of feelings of ontological features. The tree SOT (Sentiment Ontology Tree) is constructed to represent the features of an object hierarchy. Each node the tree contains as children. Besides these features there are two leaf nodes representing the negative and positive feelings of the feature represented by the node. The classification approach used is based on hierarchical classification algorithm. The algorithm takes as input a SOT and texts already sorted and aims to validate the hierarchical construction of sentimental texts. The results demonstrated that knowledge of hierarchical relations improve performance and accuracy of sentiment analysis. In addition, you can use a generic model with a SOT composite, SOT of individual objects, and a root node. This adjustment allows the algorithm to be used with general texts (i.e. not containing a predefined object).

[Neviarouskaya et al. 2011] article in The Lexicon for Sentiment Analysis describes a method for automating the generation and marking values for level of feeling subjective text fragments called SentiFul. The idea is to enable any basis to expand through techniques such as direct synonymy, antonym, relations of exploitation, hyponymy derivation, and composition, among others. The proposal is made pursuant to textual recognition, using four types of affixes (used in the derivation of new words), depending on their role with regard to feelings such as propagation, reversal, intensification, and weakening. The derivation is done to find new words using such composition. This process generates a large number of terms useful especially in the case of nouns and adjectives. The algorithm is designed for the automatic extraction of words related to sentiment using terms from WordNet (but using words from SentiFul).

3. Tools and methods

This section presents the tools and methods that were used in the analysis process of feeling as well as a detailed description of the process developed and proposed in this research.

3.1. Tools for the process of sentiment analysis

Below the software that makes the PALAVRAS software which performs the semantic analysis of text and was used in the development of the process and the creation of the feelings dictionary is described.

3.2. The Palavras software

The process developed in the research consisted primarily of semantically analyzing fragments of texts (articles and reviews) of social networks in order to extract information from feelings. To do a semantic analysis of text using WORDS software (developed by Eckhard Bick and based on corpus "Syntactic Forest" of Linguateca) was performed. This is an automatic parser for Portuguese that performs parsing, syntax analysis of the Portuguese language and is able to provide morphological information of a sentence.

The process of sentiment analysis prepared by the research begins with the use of WORDS as parser and lexicon. This software is used as the basis of the algorithm, "normalizing" the input and parser. The process of sentiment analysis begins with the extraction of text elements related to the view, then uses the classification of opinion as to his character considered within the scope of positive, negative or neutral. The sequence is performed to compare their opinions and judgments, and commonly uses the term "object" to refer to the target's opinion, which may contain several features or subparts. These may also be subject to reviews.

3.3. The sentiment dictionary used

The construction of the dictionary of feelings was based on the classification of feelings dictionary Sentistrenght [Thelwall et al. 2012], a dictionary with tens of thousands of terms denoting feelings. The purpose of the dictionary is to quantify feelings. A sample entry in this dictionary would be: "bad: -2", which means that the English word carries a negative feeling with numeric value -2. According to the authors, the dictionary was constructed from research in psychology, philosophy and linguistics. The biggest challenge for the use of the dictionary SentiStrenght during the project was to undertake the translation process while maintaining the real meaning of the words in the English language. The process of translation dictionary Sentistrenght that contains a list of approximately over 22,000 words, through a process of semi-automatic translation, was divided into three steps described below. The first step comprises the initial translation.

We used three tools: the Bing Translator from Microsoft, Google and Yahoo BabelFish Translator. From the translation made by each tool, an index of agreement was created. With this it was possible to filter terms with higher disagreement among dictionaries and therefore need more attention. The second step consists in validating the translation of terms. Despite the undoubted utility of translators, automated much texture characteristics of each tongue are not detected, so that often the manual intervention of a person skilled in translation is necessary. As it would be very costly to allocate an expert to translate all the terms in question, it was decided to automate the process. The process consists in the execution of a program designed to access the COMPARA, which is a parallel corpus English/Portuguese available from Linguateca (distributed resource center for the Portuguese language). The operation of COMPARA is as follows: given a term in English (or Portuguese), he shows us how it was the translation of that term in several different contexts, including works of Machado de Assis, Eca de Queiros and Aluisio Azevedo.

The program developed at this stage serves as a crawler, referring COMPARA for each of the search terms and registering cases where translations are relevant and where there is no match. Thus, we validate the suggested translation of automated translators from translations made by professionals. The third step was termination. At this stage we select the most relevant terms (i.e., terms that would result in inaccurate translations and have greater negative impact in the search results). With these terms, an inspection was made in each of the translations, looking for imperfections in the translation process done so far. In this step, just under 600 terms were analyzed.

3.4. Domain-specific ontologies

After defining the analysis of texts and the process of translation dictionaries of terms related to feelings that are considered in the research, we moved to the step of defining ontologies. The concept of ontology was used to guide the process of identifying objects (in this case car models of Fiat and non-FIAT), characteristics (properties of objects such as power, beauty, etc.) , contexts (following paragraphs of opinions on the same object). These are also vital in the final classification. For the development of this research, we used the concept of ontology used in the field of information science as a formal knowledge, a set of concepts, and their relationships in a domain. Ontologies can be used to model human and abstract concepts. The formalization allows systems to take advantage of human models. Ontologies were created based on models of FIAT. The first ontology of "FIAT models" used in this research adds the necessary information about the Fiat car models. This ontology allows us to not only detect objects of interest in the text (e.g. "Fiat Palio") but also to determine the class that the given object belongs to (a car of the "Hatch" subtype or "Hatch Compact" as shown in Figure 1.

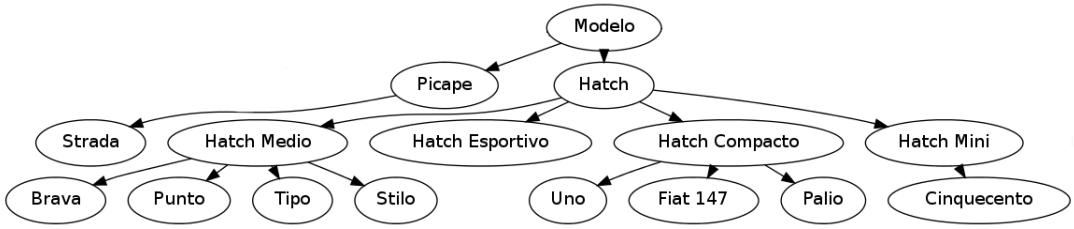


Figure 1: Example of part of the ontology: FIAT Models

The second ontology "Car Features" organizes the relevant features such as an automobile, or adds components and features that are targets of feelings (Figure 2).

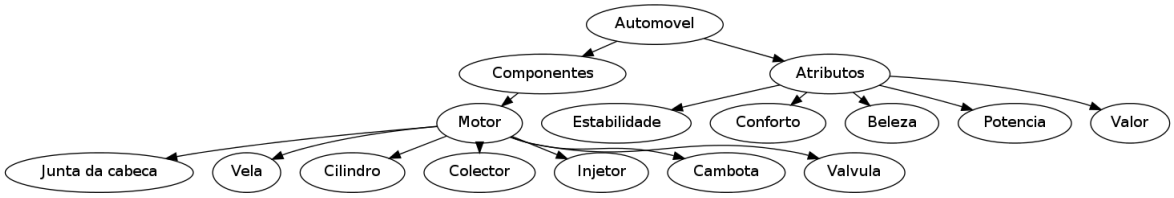


Figure 2: Example of part of the ontology: Features of a car

The third ontology "Non-FIAT Models" is made up of other objects (cars) that have the same features (features and components) of the objects of interest. The features associated with these non-FIAT objects must be detected and correctly excluded from the analysis because it would cause distortion of the result.

3.5. Complete overview of the analysis process

After the definition of ontologies the process of sentiment analysis proposed was implemented in a prototype. Generally, the process consists in the capture, analysis and storage of opinions. More specifically, the process is divided into eight stages that come from the collection of opinions on social networks to the aggregation of the opinions rendered, as shown in Figure 3. The first steps to represent the views containing the text is captured and standardized. Then the objects of interest (defined by the ontology) are found in the text, as well as their features (characteristics). Subsequently, the detection and the calculation (based on dictionary feelings) of sentiment related to each of the objects of interest and its features are done. Finally, the results are analyzed and stored in forms of reports.

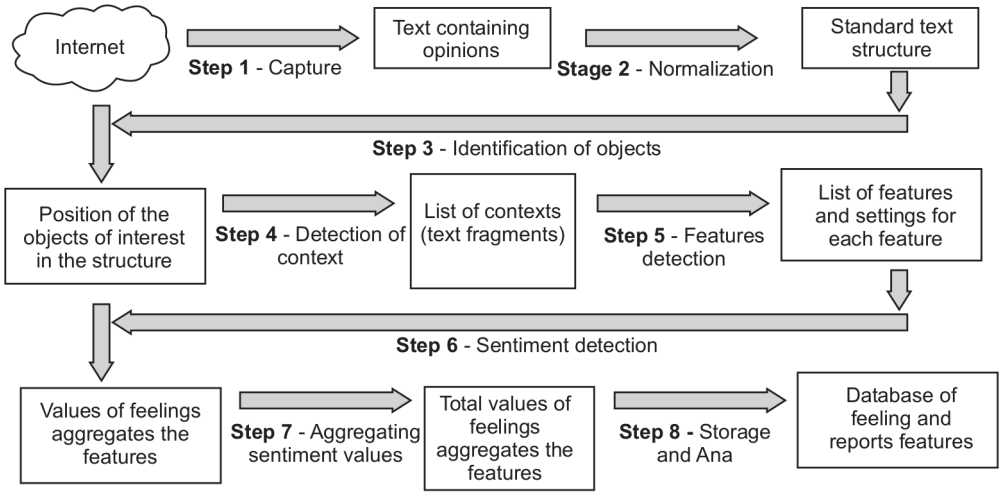


Figure 3: Overview of the review process.

3.6. Step 1 - Capture

Initially, several blogs themed "cars", which were openly discussed among readers were accessed by a crawler to build the basis of texts (corpus analysis). A funding program was developed to follow accompany the new articles and news sites selected, via RSS (RDF Site Summary). The capture process started by selecting RSS feeds (RSS news aggregator) from sites of interest, such as portals and blogs with the car theme. Then the inner content is extracted and stored as raw XML data and HTML (RSS formats in which are stored on the web). In the second step these raw data are processed by separating the HTML and XML texts of interest to the news and any comments. The result of this step is a collection of texts grouped as news/article (title and text) and the texts of any related comments. This corpus comprises the input for normalization algorithm.

3.7. Stage 2 - Normalization

After the texts were captured, it was necessary to normalize and prepare themfor algorithm review. In order to this, it is necessary to obtain the morphological and lexical structure of the texts, and bring all terms to their most basic form (infinitive). This step is performed by PALAVRAS software. Below is an example of normalization using the words in a text captured (Figure 4): "I do not know if it's the best, but I am very pleased with my Palio. It is economical and never gave me problems and I have it already since 2003." Text taken from `http://br.answers.yahoo.com/question/index?qid=20061112175518AAfs7a3.`

não [não] **ADV** @ADVL>
sei [saber] <vt> <fmc> **V** PR 1S IND VFIN
@FMV
se [se] **KS** @SUB **@#FS-<ACC**
é [ser] <vK> **V** PR 3S IND VFIN @FMV
o [o] <artd> **DET** M S @>N
melhor [bom] <KOMP> <SUP> <n> **ADJ** M
S @<SC
,
mas [mas] **KC** @CO
estou [estar] <vK> <fmc> **V** PR 1S IND
VFIN @FMV
muito [muito] <quant> **ADV** @>A
satisfeito [satisfazer] <vt> **V** PCP M S
@<SC
com [com] **PRP** @<ADVL
o [o] <artd> **DET** M S @>N
meu [meu] <poss 1S> **DET** M S @>N
pálio [pálio] **N** M S @P<
.

Figure 4: Normalization output.

3.8. Step 3 - Identification of objects.

Along with the standardized texts, in this step the ontologies of the Fiat brand cars (besides the very word FIAT) and the ontology of brands "Non-FIAT" is used to perform the detection of objects of interest. This step results in the positions of words in texts identified as objects of interest, being the objects of the models identified both FIAT cars (car models identified by Fiat) and non-FIAT (models from other companies) in the texts, we call these positions simply markers of objects that represent the position of words in the text, identified as objects of interest (car models). Recording the positioning is done by counting the number of words required to reach the word object as the beginning of the phrase or the piece of text, that is, the first word reading the text position is zero, the second is a position, and so on.

For example, the text is the sentence: "I prefer the Corsa, it is softer and more comfortable. The Palio is also good, and has more interior space. But the Gol, I think is very hard." removal of `http://br.answers.yahoo.com/question/index?qid=20070227083007AAf9lKv`. This text contains an object of interest present in the FIAT ontology (Palio object) at position 10, and two other objects of the Non-FIAT ontology (the Corsa objects at position 3 and Gol position number 21).

3.9. Step 4 - Detection of context

The next step consists of extracting the contexts of the objects detected in the previous step. A context, in the proposed process, is represented by a piece of text. Thus, the text of the sentence example of Step 3 is divided into contexts, as illustrated below (Figure 5).

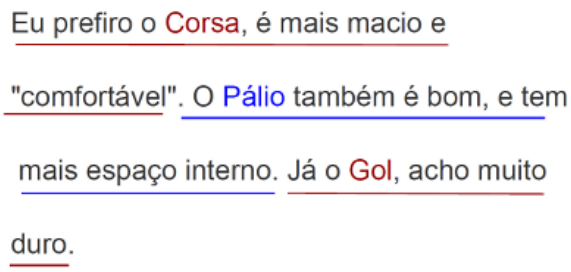


Figure 5: Context detection. "I prefer the Corsa, is softer and more comfortable. The Palio is also good, and has more interior space. I think the Goal is too cumbersome."

3.10. Step 5 - Features detection

Separate contexts, the next step in the detection takes place in the nature or subparts (features) of the car in each of the contexts generated in the previous step. In this case we, use the ontology features of cars (Figure 2) to guide the process (Figure 6).

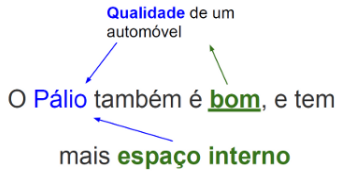


Figure 6: Features detection.

3.11. Step 6 - Sentiment detection

After detecting the features, the sentiment detection process is carried out. At this stage, SentiStrenght is used to detect and classify the sentiment level. Feelings are related features closer (Figure 7).

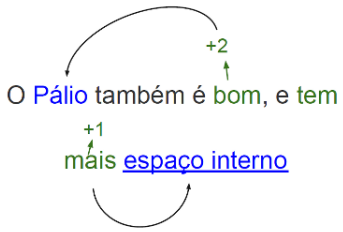


Figure 7: Sentiment detection.

3.12. Step 7 - Aggregating sentiment values

Then it is made of an aggregate structure of the detected feelings ontology feature cars (Figure 2). It can therefore be inferred, for example, if somebody speaks well about the power of a car, he/she is referring indirectly to the motor of the car. When checking the feeling at any point on the ontology tree , the feeling values aggregated to the descendants at that point are also recorded for the parent, as well as for all the other points up to it. the feelings of the current point. Therefore a positive feeling about the tires of a car model, is automatically recorded to the car itself reaching the car brand car that gets all the sentiment at the root of the ontology tree.

3.13. Step 8: Storage and Analysis

At this stage of feeling all the information is stored, keeping references to the car model (FIAT ontology), feature (cars ontology) and frameworks for future validations. This allows various types of cross-references and comparisons.

4. Results

The first result obtained in this project was the construction of the ontology of objects of interest of Fiat cars, the brand researched, the nonFiat objects, and components of cars. These ontologies represent concepts and relationships within the domain of the car market. From these data models can make inferences about the domain objects. Were used both for identifying our target objects, as a bag of words (*BOW*) initially and latter as a sentiment holder linking objects to sentiments. This allowed the system to measure sentiments at any level of the ontology structure by summarizing the sentiment assigned to descendant objects.

Our second result, in a smaller scale, is our sentiment dictionary. Our methodology was successful in translating this type of dictionary from another language without losing too much consistency and meaning while keeping the sentiment strength values. Although some human labor was needed to check the word list coherence most of the process was automatized and could be applied to other languages as well that lacks or have little availability to a proper sentiment dictionary.

The proposed methodology to extract, process and analyze sentiments and opinions on the Web, by using a unsupervised algorithm and the resulting prototype created by implementing this methodology are our third and most important result. When properly loaded with processed sentiment and relationship information, ontology trees can provide a dynamic yet simple way to relate, summarize and visualize processed data. Based on that information, many types of reports can be generated without the need of reprocessing data or sentiments. Comparative studies are a very good example of this concept. With a loaded ontology tree, important questions can be answered such as “Which car is better, based on customer feedback, Corsa or Palio?”. The algorithm just needs to summarize every sentiment related to Palio and Corsa, and the decedents. But if the question is ‘Which car is considered prettier?”, then just sentiments related to each car and its appearance (remembering that appearance is an ontology object defined as *part of* a vehicle) are used. Many different views on the ontology that answers countless meaningful questions like those on consumer opinions, can be generated.

Figure 8 shows our results on the test made partially implementing our methodology that can serve as an example of the report output result that can be obtained from the prototype. Keep in mind that this is a preliminary result, just to exemplify what kind of report can be made using our methodology. This was run as a profofconcept to test if our methodology could process a simple dataset and provide a useful (even if limited) report. We expect much more detailed tests in the following months.

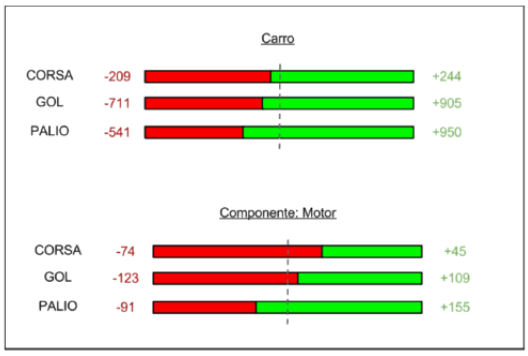


Figure 8: Partial results.

In this particular case (Figure 8 results), opinions were extracted only concerning *Corsa*, *Gol* and *Palio*, which are car brands in direct competition in Brazil. The dataset was obtained from open articles found in social media on the Web mainly constituted by news feeds from many automotive magazines sites, blogs and discussion groups. This dataset comprised 8643, articles with 83.571 related comments, for a total of 607.8527 words collected in 53.4846 sentences written by 4.112 authors (combining articles and comments data). Based on the results obtained it can easily be inferred that there are more positive than negative opinions about Palio: 63% of the opinions analyzed were considered positive while 37% were considered negative. For the brand Gol, 47% of the opinions were positive, while 53% were negative. Finally Corsa 38% positive opinions while 62% were negative for the prototype. It can be seen that, while Gol showed a more mixed public opinion result, having almost the same amount of positive and negative opinions, Palio had a more positive public opinion Corse a more negative one.

The validation of results for this test run is still ongoing and, since it had to be done by hand, only a very small set of data (only 24 articles and 63 comments) until now. This represents yet a too small validation information to provide meaningful confidence intervals. However in the set already validated we were able to perceive small number of wrong sentiment values or false positive ones (less than 6%) but, a considerably number of false negatives or sentiments not recognized (close to 15%) that we believe can be attributed to the very simplistic and small ontology trees that were used in this first test.

5. Concluding remarks

This paper presented a methodology to process, analyze and summarize sentiments and opinions from sentences extracted from the Web. This methodology was applied to the automotive field for specific analysis of the FIAT brand, resulting in a prototype that is still incomplete, but which could easily be applied to any other domain. Although the practical results were very simple, especially considering the potential for the methodology, it was sufficiently successful to serve as a proofofconcept that the methodology works and can provide interesting insights about the data.

Our practical result, on Figure 8 is an illustration of how we can visualize results from the methodology. As our prototype is in its early stages of development this result is just a proof of concept as only a small dataset was processed for just 3 objects of the ontology trees: Corsa, Palio and Gol brands of vehicles. As the development progresses we will be able to show much more detailed insights of much larger datasets. In this

particular result, we showed that the Palio brand received a much more positive sentiment value then negative, while Corsa brand received the opposite results and Gol opinions were more mixed.

In a recent conference the group was asked about the possibility of changing the process described in Section 3 to make it able to collect information about the author's identity details (sex, age, country) and if the reports could be placed on a timeline, so that analysis could also identify trends. These ideas were noted as possible future developments since it could enrich the scope of the methodology described here. Also as future work we intend to generalize the proposed process so it could be even more easily applied to other areas and organizations interested in monitoring opinions expressed about products and services. A first idea to implement this generalization would be using a SOT tree technique proposed by [Wei and Gulla 2010] in lieu of our simplistic ontology tree.

References

Cicortas, A., Iordan, V., and Fortis, A. (2009). Considerations on construction ontologies. *CoRR*, abs/0905.4601.

Kunze, M. and Rösner, D. (2005). Context related derivation of word senses. *CoRR*, abs/cs/0501095.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2nd ed.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2011). Sentiful: A lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 2:22–36.

Polpinij, J. and Ghose, A. K. (2008). An ontology-based sentiment classification methodology for online consumer reviews. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pages 518–524, Washington, DC, USA. IEEE Computer Society.

Ramanathan, J. and Ramnath, R. (2010). Context-assisted sentiment analysis. In *The 25th Annual ACM Symposium on Applied Computing*, pages 404–413, Sierre, Switzerland. ACM.

Rösner, D. and Kunze, M. (2003). Exploiting sublanguage and domain characteristics in a bootstrapping approach to lexicon and ontology creation. *CoRR*, cs.CL/0304035.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *JASIST*, 63(1):163–173.

Wang, S., Li, D., Song, X., Wei, Y., and Li, H. (2011). A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. *Expert Syst. Appl.*, 38(7):8696–8702.

Wei, W. and Gulla, J. A. (2010). Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 404–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

A Foundational Ontology to Support Scientific Experiments

Sergio Manuel Serra da Cruz[1,3], Maria Luiza Machado Campos[2], Marta Mattoso[1]

[1] Programa de Engenharia de Sistemas e Computação (PESC/COPPE-UFRJ)

[2] Programa de Pós Graduação em Informática (PPGI-UFRJ)

[3] Programa de Pós Graduação em Modelagem Matemática e Computacional (PPGMMC - UFRRJ)

serra@ufrrj.br, mluiza@ufrj.br, marta@cos.ufrj.br

Abstract. *Provenance is a term used to describe the history, lineage or origins of a piece of data. In scientific experiments that are computationally intensive the data resources are produced in large-scale. Thus, as more scientific data are produced the importance of tracking and sharing its metadata grows. Therefore, it is desirable to make it easy to access, share, reuse, integrate and reason. To address these requirements ontologies can be of use to encode expectations and agreements concerning provenance metadata reuse and integration. In this paper, we present a well-founded provenance ontology named Open proVenance Ontology (OvO) which takes inspiration on three theories: the lifecycle of in silico scientific experiments, the Open Provenance Model (OPM) and the Unified Foundational Ontology (UFO). OvO may act as a reference conceptual model that can be used by researchers to explore the semantics of provenance metadata.*

1. Introduction

Currently, researchers are facing a proliferation of a huge volume of scientific data. These data are often shared and further processed and analyzed among collaborators. There is a consensus among science data communities that metadata is the foundation for data discovery, use, and preservation. The importance of managing the provenance metadata of scientific experiments is becoming vital to researchers as they have to share results and also consider the long-term usability of data products generated by their investigations.

Provenance captures a derivation history of data products, and is essential to the long-term preservation, to reuse, and to determine data quality (Freire *et al.*, 2008, Moreau *et al.*, 2011). Provenance provides transparency in data acquisition and processing, managing trustworthiness of data sources, allowing those who use the data to determine its validity and to verify its accuracy. For instance, in scientific domains such as Biology, Chemistry and Physics, the tendency toward collaborative scientific process is increasingly evident (Hey, Tansley, Tolle, 2009). Thus, with the proliferation and sharing of such data, questions such as "where did this data come from?", "who else is using this data?" and "for what purpose was it generated?" are becoming increasingly common in the scientific arena. To ensure that data provided by third-party can be trusted, shared and reused appropriately, it is imperative that the semantics of provenance is clearly and precisely defined and made available to users.

Despite of the amount of research papers and surveys about provenance (Buneman *et al.*, 2001, Oinn *et al.*, 2004, Sahoo *et al.*, 2008, Cruz *et al.*, 2009, Mattoso *et al.*, 2010), each work describes it according to a different and particular perspective. For instance, Buneman *et al.* (2001) and Oinn et al.(2004) describe provenance in terms of common data, while others describe it in terms of metadata (Cruz *et al.*, 2009 and Mattoso *et al.*, 2010)),

i.e., as a secondary piece of information that is complementary in some way to the primary piece of information to which it refers. Unfortunately, despite of all these research efforts, scientific data and provenance integration is a problem not completely solved, especially when it involves semantic issues. With such issues in mind, we are interested in the semantics of provenance at a novel perspective when compared to other related works (Salayandia *et al.,* 2006, McGuinness *et al.,* 2007, Sahoo *et al*., 2008, Zhao, 2010).

The goal of this article is to present and discuss the features of a novel ontology named *Open proVenance Ontology* (OvO) which is inspired in three other theories: the lifecycle of scientific experiments, presented by Mattoso *et al.* (2010), the Open Provenance Model (OPM), discussed by Moreau *et al.* (2011) and the Unified Foundational Ontology (UFO), proposed by Guizzardi (2005). We advocate that when binding higher levels of provenance metadata (regarding organization and knowledge about the experiment and its scientific hypothesis) with fine grained operational provenance (collected during experiments´ execution) and the explicit specification of the conceptualizations of the *in silico* scientific experiments domain; unanswered research questions might be solved by exploring the semantics of provenance metadata. For instance, one can perform reasoning about the history of how hypothesis, models, decisions, annotations evolve during recurrent runs of a workflow that is part of an *in silico* scientific experiment. Furthermore, one can explore how an abstract workflow conceived by a researcher is related to a given data product generated at the laboratory of a research partner.

Differently from related works, in this article we present the conceptual model of a well-founded provenance ontology about *in silico* experiments. The purpose of the ontology is to provide a provenance reference model in order to: (i) support the integration of distinct kinds of provenance produced during the lifecycle of scientific experiments; (ii) address semantic interoperability and data/standard integration between experiments; (iii) allow a novel approach for provenance querying; (iv) convey a knowledge repository about scientific experiments results; and, finally, (v) represent the provenance of scientific experiments as a semantic network, exposing it to automated capture and query mechanisms.

The remainder of this paper is organized as follows: Section 2 provides a brief background about the role of provenance in the lifecycle of a scientific experiment; Section 3 introduces the ontological engineering approach adopted in this work; Section 4 presents the OvO ontology; Section 5 discusses related work; and, finally, Section 6 concludes the paper.

2. Background – The role of provenance on *in silico* scientific experiments

In silico is an expression used to mean "performed on computer or via computer simulation". *In silico* scientific experiments are characterized by the composition and execution of several variations of scientific workflows (Mattoso *et al.,* 2009); they are performed with the aid of computer systems and provide researchers a number of advantages, such as: higher precision and better quality of experimental data; better support for data-intensive research and access to vast sets of experimental data generated by scientific communities; more accurate simulations through scientific workflows and higher productivity. However, *in silico* experiments nowadays suffer from an increasing complexity on setting up, maintaining and making changes to the simulation systems and also the shortcomings of managing large data sets of experimental data and provenance metadata.

Like traditional scientific experiments, *in silico* scientific experiments, independent of their domain, follow some common directions (Jarrad, 2001) such as: (i) they need to be

re-executed, enabling other researchers to conduct similar experiments to confirm (or refute) the results; (ii) the results need to be well documented to be used as a baseline for further experiments, conducted by different researchers in different laboratories; (iii) they must follow a protocol or a methodology and be executed under controlled conditions. However, *in silico* experiments are often specified and materialized as scientific workflows.

Scientific workflows are defined in two levels. In a higher level of abstraction, an abstract workflow is described as conceptual model without binding to specific computational resources. In the lower level, a concrete workflow binds programs and data allowing the structured composition of a sequence of operations aiming its execution to achieve a desired scientific result. To be effectively managed the scientific workflows require a specific set of cardinal facilities, such as experiment specification techniques, workflow derivation heuristics, provenance gathering mechanisms and high performance computing environments ranging from private clouds, commercial clouds such as Amazon, GoGrid, Rackspace to supercomputing centers (Stevens *et al.*, 2007, Mattoso *et al.*, 2010).

Manual analysis of the results data set of *in silico* experiments is commonly unfeasible. This involves, for instance, checking input and output data sets of each activity of the workflow, verifying if computations failed on remote computational resources, and checking all activities that contributed to the creation of a particular data set. Many of these activities can be automatically executed by querying provenance metadata.

The treatment of provenance as a first-class data artifact was primarily introduced at OPM by Moreau et al. (2011). The OPM is a way of recording information about artifacts, agents and processes as they occur which includes constructs for representing causal and dependency relationships between sub-processes and the data items or other artifacts that they use or produce. The OPM is a provenance metamodel which is gaining popularity and for which implementations are becoming available in OWL, RDF and Java. Despite of its increasing popularity OPM has some issues, for example, it neither considers all kinds of provenance of in silico experiments nor expresses an unambiguous semantic model.

There are two types of provenance. *Prospective provenance* describes how these scientific workflows were composed and *Retrospective provenance* describes how they were executed and also the relationships between the input and output data sets (Freire *et al.*, 2008). Another important characteristic of provenance is related with its granularity (Cruz *et al.*, 2009), also referred to as provenance level. It is generally classified as coarse or fine-grained. The desirable level of provenance granularity depends on application domain requirements, and the cost of collection, storage and processing. The finer the granularity of the provenance record, the higher the information entropy and associated cost.

According to Mattoso *et al.* (2010), *in silico* scientific experiments can be described as being part of a lifecycle, which consists of three stages: *Composition, Execution* and *Analysis*. Each stage has an independent cycle, taking place at distinct moments of time and handling different kinds of provenance metadata. At the *Composition* stage, researchers either elicit the requirements to build a new abstract workflow as software or retrieve old ones to reuse for new purposes. They start the composition process building the raw version of the workflow by choosing programs incrementally, backward or forward, along the stage. Then, they refine the successive versions towards a concrete workflow.

At the *Execution* stage researchers make their essays by executing instances of a concrete workflow according to their own needs using real parameters and data sets in a production environment. Researchers can also monitor (local or distributed) experiments and register retrospective provenance metadata that can be further used in debugging or

optimization activities, e.g., researchers may optimize concrete workflows, this can be obtained by usage of parallelism and distributed computation. They are also able to initiate an analytical process undertaken over outcomes of collections of experiment runs,

Finally, at the *Analysis* stage researchers can investigate the data products generated by the experiment and then publish results or share not only its outcomes, but also the whole workflow or its parts to other domain experts. This stage may be further decomposed to support the analysis of results. The researchers may face two different situations when analyzing the results: (i) with the generated data, they may conclude that results are likely to be correct, but decide to continue with the experiment, varying parameters and ingesting others data sets, to prove or refute the hypothesis; (ii) when analyzing the data products, researchers discover that their hypothesis is refuted, but faces a new scenario that may lead to a new discover, thus raising a new hypothesis.

In this article we advocate that *in silico* scientific experiments can be fully described as hierarchical trails of provenance metadata with different kinds of granularity collected during its lifecycle. First, at a higher and abstract level, we have the prospective provenance; it cannot be gathered by the existing Scientific Workflow Management Systems (SWfMS); at a lower level we have fine grained provenance collected during the execution and analysis of the experiments.

3. The Ontological Engineering Approach

The first version of the Open proVenance Ontology (OvO) was initially developed by Cruz (2011). The author adopted a combination of two methodologies found in the literature. Firstly, we have employed the ontology engineering process which includes the phases of conceptual modeling, design and codification (Gomez-Perez *et al.,* 2004 and Guizzardi, 2007). The conceptual modeling of ontologies should strive for expressivity, clarity and truthfulness in representing the subject domain at hand. Due to space restrictions, the methodology and the rationale behind of the OvO´s conceptual modeling and design will not be fully discussed in this paper. A detailed description can be found at Cruz (2011).

As second methodology, we have used the ontologically well-founded UML modeling profile proposed by Guizzardi and Wagner (2005). This profile comprises a number of stereotyped classes and relations implanting a metamodel that reflect the structure and axiomatization of a foundational and domain independent ontology named Unified Foundation Ontology (UFO). The UFO was stratified in three ontological layers (fragments), namely UFO-A, UFO-B and UFO-C. A complete description of UFO falls completely outside the scope of this paper. However, we give an overview of the fragments of this ontology which were used in the instances of the modeling profile employed in this article.

UFO-A is the core of UFO, it defines concepts related to *endurants.* An endurant is an entity that is present as a whole at any given point in time, i.e., it does not have temporal parts and persists in time while keeping its unique identity. The endurants (universals and particulars) can be summarized as follows: *Kinds* are rigid, independent sortals that supply a principle of identity for their instances; *Phases* are independent anti-rigid sortals; *Roles* are anti-rigid and relationally dependent sortals, *RoleMixins* are non-sortals that can be partitioned into disjoint subtypes which are sortals. Examples of endurants are a laboratory, an organization, a researcher, an experiment and its phases. UFO-A also uses relations. Relations are entities that connect other entities together, they are divided as follows: Formal relations hold between two or more entities directly, without any further intervening individual. Examples of formal relations include “a laboratory has projects”. Contrarily, material relations have a material structure of their own and have their *relata* mediated by

individuals called relators. For instance, a workflow_run is a relator that connects a concrete workflow and an executor.

UFO-B builds upon UFO-A and defines concepts related to *perdurants*. A perdurant is an entity composed of temporal parts, i.e., its existence extends in time accumulating temporal parts. Examples of perdurants are the codification of a concrete workflow and the composition of an abstract workflow. It is not always the case that whenever a perdurant is present all of its temporal parts are also present. With a perdurant, if we freeze time we can only see a limited number of parts of the perdurant and not the whole. For instance, in a "coding session" of an concrete workflow, if we take a snapshot at the point in time when the programmer is having its code validated in order to execute a concrete workflow, we cannot determine that this task is part of the "execution" of the scientific experiment.

UFO-C defines social and intention-related concepts (both *endurants* and *perdurants*) and is built on top of UFO-A and UFO-B. One of the main distinctions made in UFO-C is between *Agents* and *Objects*. An Agent is a substantial that creates actions, perceives events and to which we can ascribe mental states (intentional moments). Agents can be physical (a football player) or social (a stadium). An Object is a substantial unable to perceive events or to have intentional moments. Objects can also be further categorized into physical (e.g. a ball) and social objects (e.g., money).

4. Open Provenance Ontology[1]

In this section we discuss the key concepts of Open Provenance Ontology (OvO). OvO´s concepts are depicted as UML class diagrams because of the widespread understanding of UML classes and relations and their suitability for our purposes of illustrating how the presented concepts are organized and how they are related to each other. OvO was developed as a set of three sub-ontologies: (i) *in silico scientific experiment sub-ontology*, (ii) *experiment composition sub-ontology*, (iii) *experiment execution sub-ontology*. The sub-ontologies complement each other; they are connected by relations between their concepts as well as by formal axioms.

The UFO stereotypes of the modeling profile used are signed between the sign << >> and the names of the concepts of the ontology (classes) are typed in *italics*. The classes are colored to facilitate understanding. The concepts colored in yellow map prospective provenance and in green retrospective provenance. Due to space restrictions, solely the key concepts are discussed, the complete description of all concepts of OvO can be found at (Cruz *et al.*, 2009 and Cruz, 2011).

4.1. The *in silico* scientific experiment sub-ontology

This sub-ontology captures the structure of *in silico* scientific experiments at a high abstraction level (Figure 1).

An *in silico* experiment is a research that has the purpose of discovering something unknown by adopting a computational model and by the evaluation of a hypothesis that involves the design and implementation of an *in silico* experiment. *Experiment* and *Project* are labeled as <<Kind>> that is a rigid sortal universal that can be identified in all possible worlds. We consider *in silico* experiments as a composition of three essential parts: the model, the hypothesis and project (all are represented by the *essential=true* tagged value in the part-whole relation notion); this relationship indicates that an experiment is made of parts and inseparable relationally dependent. *Hypothesis* and *Model* are also rigid sortal

[1] Only key concepts and relationships were illustrated and discussed.

universals, represented as disjoint concepts tagged as <<Kind>>. The models do not exist outside the context of the experiment as a whole. They are inseparable parts of the scientific experiment they comprise (represented with the mark *inseparable=true*). The *Model* defines the limits of a scientific investigation, recording the circumstances surrounding an experimental study. The *Hypothesis* is a proposition that is stated in an attempt to verify the validity of a provisional response.

Project refers to the research project in which the *in silico* experiments are conducted. A project defines the location of the research and the involvement of several types of researchers. *Project* is represented as an hierarchy. The *Laboratory* and *Organization* concepts are rigid sortals and are related by aggregation and composition with the underlying concepts. The *Laboratory* uniquely identifies the points in geographical space where a research project is designed and where the *in silico* experiments are conceived.

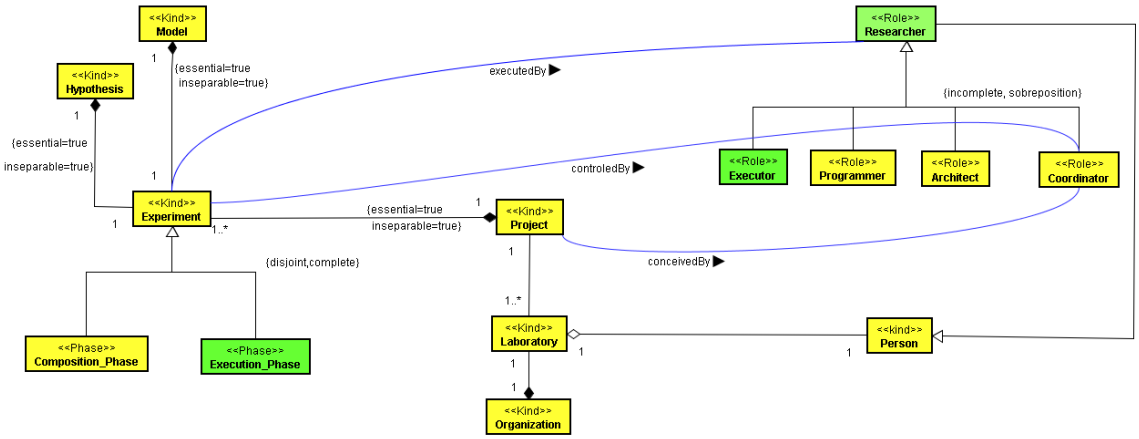


Figure 1. Fragment of the *in silico* scientific experiment sub-ontology

The *Organization* is a rigid sortal universal which plays an important role in the sub-ontology. For example, COPPE/UFRJ is an organization that has several laboratories (*e.g.*, databases, software engineering, among others) and they have material and human resources that can be allocated to conduct a research project. *Laboratory* and *Organization* are independent, but complementary, one can change the members of an organization (university, laboratory) without losing its identity principle. The *Laboratory* also has another important feature, it stresses the relationship with *Person* as an aggregation (tagged as a rigid sortal). Here, we assume that being a researcher is an extrinsic property of a person, *i.e*, there are worlds in which a person is not a researcher and however, he still remains a person. *Person* is regarded to any human being, however, only those who plays any role in conducting a scientific experiment are mapped as *Researcher*. To represent the possible roles played by people whose main attribute is to be a researcher, we adopt the stereotype <<Role>> of the UFO-A, representing them as anti-rigid and relationally dependent sortals. Each instance of researcher must be an instance of person who inherits its identity principle. The concepts *Executor*, *Programmer*, *Architect* and *Coordinator* are sub-types of *Researcher*.

Returning to *Experiment*, it is tagged as rigid sortal and its graphical representation is the same as a generalization of the UML metamodel. However, we can notice different semantics. For instance, in UML, the classes to a generalization are necessarily disjoint and by default do not form a partition. Taking into account the lifecycle of *in silico* experiments (as described in Section 2) and the theory behind UFO-A. An experiment may be represented as a disjoint set (label *{disjoint, complete}*) of stages. It can be decomposed by

two different concepts: *Composition_phase* and *Execution_phase*. Both are tagged as <<Phase>> which means that each stage of the lifecycle is an independent anti-rigid sortal that happens in different points in time and represents a condition that depends solely on its intrinsic properties.

4.2. The experiment composition sub-ontology

This sub-ontology represents *prospective provenance* associated to the composition stage of an *in silico* experiment (Figure 2). The concepts related to prospective provenance are yellow colored while the ones related to retrospective provenance are green.

The composition stage of *in silico* experiments comprises all tasks of specification and modeling of abstract and concrete workflows and their activities. During the modeling task, we capture the knowledge related to the materialization of the experiment and the design of the scientific workflows. At the highest level of the metamodel there is the concept *Composition_Phase*. Such concept is tagged as anti-rigid sortal. By this kind of representation we ensure the unique identification of each cycle of composition in the life cycle of the experiment executed in a given organization.

The association between *Composition_Phase* and *Workflow* has an important semantic explanation; it allows the specialization of the two types of workflows found on *in silico* experiments: abstract and concrete workflows. *Workflow* is tagged as <<Complex event>>, which means that such kind of event is composed of other events by means of event composition operators.

UFO-B events are possible transformations from a portion of reality to another, i.e., they may change the reality by changing the *state of affairs* from one (*pre-state*) situation to a (*poststate*) situation. These are complex entities that are constituted by possibly many endurants. Situations are taken to be synonymous to what is named *state of affairs* in the literature, *i.e.,* a portion of reality which can be comprehended as a whole. According to our metamodel, the concepts *Workflow_Code* and *Workflow_Description* are tagged as <<Kind>>, such approach allow us to uniquely identify the versions from one prior-version of a workflow (*pre-state*) to a novel-version (*poststate*) of a workflow.

Complex events are aggregations of at least two perdurants (either atomic or complex events). Perdurants are ontologically dependent entities, *i.e.,* perdurants existentially depend on their participants in order to exist. The *Concrete_workflow* only exists with the participation of the *Workflow_code*, the *Concrete_activity* and the *Atomic_Concrete_Activity* (the substantial in the model). Similar arguments can be used for *Abstract_workflow*.

Concrete_activity and *Abstract_activity* are also complex events. For instance, *Concrete_activity* is modeled using the weak supplementation pattern (Guizzardi, 2005). The pattern represents a parthood where the hollow diamond is connected to the whole (*Complex_Concrete_Activity*) and the aggregation is supposed to be a irreflexive and anti-symmetric relation.

The specialization of the concept *Researcher* (discussed in section 4.1) in distinct roles, *i.e.* relationally dependent sortals, is crucial to expose the different duties of the members of a research team during the lifecycle of the experiment. For example, the architect of the experiment is a highly trained domain researcher who is in charge of planning, design and oversight/supervision of the experiment, he may not worry about the software packages versions involved, technical and operational details needed to build concrete workflows. His duties are related to the composition of the sequence of activities of an abstract workflow, while the programmer is related to technicalities and the

production of executable codes. He is the person who writes concrete workflow as computer software. The roles represented by *Architect* and *Programmer* are tagged as <<Role>> having two explicit relationships. The first refers to actions that happen in time, the actions *Coding_Session* and *Design_Session* are tagged as <<Action>>, the second relationship involves the rigid sortal *SWfMS*.

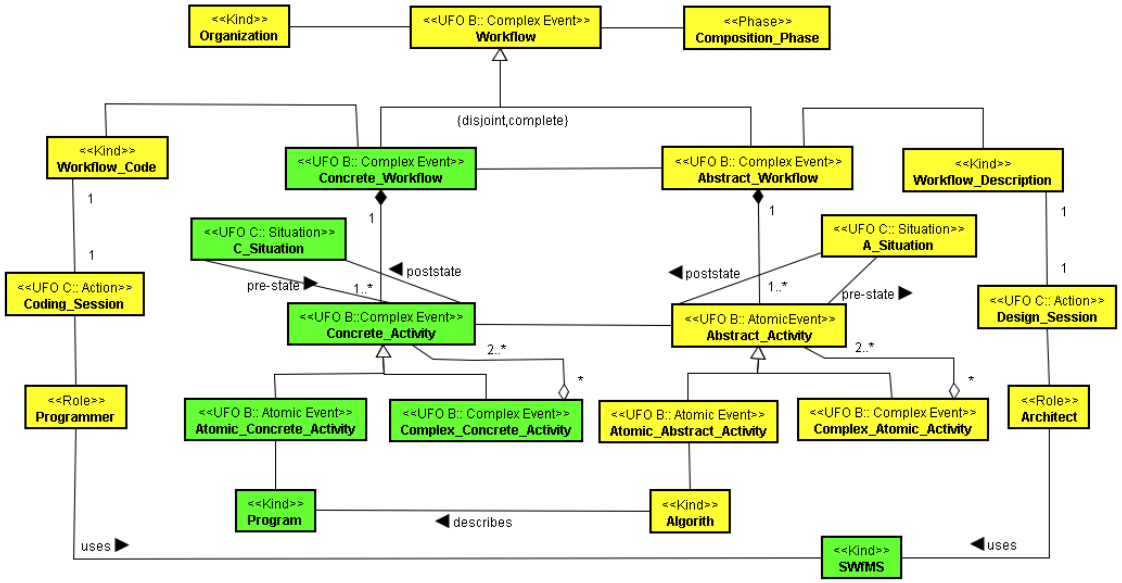


Figure 2. Fragment of the experiment composition sub-ontology

The semantics of *Coding_Session* and *Design_Session* exposes different events that happen in time (the act of coding a concrete workflow or designing an abstract workflow respectively). An *Action* is a UFO-C perdurant that is an individual instance of an *Action Universal*, with the purpose of satisfying the propositional content of an intention. Actions are intentional events, *i.e.*, events which instantiate a plan with the specific purpose of satisfying some internal commitment of entities capable of bearing intentional moments. *C_Situation* and *A-Situation* are tagged as UFO-C <<Situation>>.

4.3. The execution experiment sub-ontology

This sub-ontology represents *retrospective provenance* associated to the execution stage of an *in silico* experiment (Figure 3). The concepts related to retrospective provenance are green colored.

At the highest level there is *Execution_Phase* tagged as a universal sortal <<Phase>>; such representation ensures the unique identification to each execution of a concrete workflow during the lifecycle of the experiment. During the execution stage, the researcher can execute different instances of a concrete workflow. Therefore, the specific workflows of an experiment are associated by composition to the execution stage.

A concrete workflow comprises of one or more concrete activities. The *Concrete_Workflow*[2] represents the association, by composition, to the concept *Concrete_Activity*. A concrete activity can be understood as a codification of an executable scientific application individually (*Program*) which shall be governed by the principle of unique identification.

[2] In order to simplify the diagram of the experiment execution sub-ontology (Figure 3), the weak supplementation pattern and the *C_Situation* on *Concrete_Activity* were omitted.

The *Artifact* is a piece of data represented by a rigid sortal. For instance, a researcher Diogo uses a resource "FastaFile_TripCruzi_20122010.txt" which is owned by researcher Alberto. All artifacts, individually, must have their own principle of identity and also additional essential properties, such as name, type and date of creation, among others. The artifacts (pieces of data) handled by a specific concrete workflow, can be specialized as: *Input_Data*, *Output_Data* and *Parameter* which are shown as <<SubKind>>.

It is worth mentioning that associations between *Artifact* and *Concrete_Activity* have a particular meaning. The association *usedBy*, *generatedBy*, and *triggeredBy*, *derivedBy* (shown in blue color in Figure 3) are inherited from the OPM metamodel (Moreau *et al*., 2011). The association describes the artifacts which were consumed by a given process (concrete activity), while the second describes those that were produced during the processing of a given activity. Finally, the third represents the sequence of the specific activities of a particular concrete workflow. Strictly speaking, the associations originally defined by OPM have semantic meanings that could be better explained. For example, *usedBy* was plotted as a simple association. However, if we consider a semantically rigorous model, the association between *Concrete_Activity* and *Artifact* can be represented through a third-class R´ type material relator to explain all the details of the semantics of this relationship.

The execution of the concrete activities of a workflow is performed in a computational environment, so it is necessary to expose this association. In Figure 3, we use a combination of *Concrete_Activity* and *Environment*. However, an execution environment is not a monolithic entity, by the contrary; it is represented by an aggregation of the rigid sortals Program and Hardware. We have described the succession of resources involved in the execution of a specific workflow. In this case, Resource is tagged as a non-sortal <<Rolemixin>> type of UFO-A. That is, an abstract class that allows for specialization of other classes and subsequent identification of these resources, as *Hardware_Resource*, *Software_Resource*, and *Researcher*. We decided to use the term *Researcher* rather than, *Human Resource*, since the second term represents the entire set of employees of an organization, while the first represents the subset of employees who are trained and qualified to conduct scientific research. *Hardware_Resource* and *Software_Resource* are tagged as <<Role>>.

Finally, it is important to stress the material relations depicted in Figure 3. The explicit association between *Executor* and *Concrete_Workflow* is represented by an n-ary relationship. In this case, such relationship is mapped as *Workflow_Run*, representing sessions of work performed by the executor during the execution of a concrete model of a scientific workflow. In light of the UFO-A, universal relators are represented by the stereotype <<Relator>> and material relations are represented through associations stereotyped as <<Material>>. The dotted line relationship between a material relationship and the universal relator indicates that the relationship is derived from a material relator. The presence of the graphic symbol (●) on the relator (at the end of the dotted line) is used to distinguish the graphical representation of a class of a simple UML association class. The relationship between the relator *Workflow_Run*, *Concrete_Workflow* and the *Executor* tagged as an anti-rigid <<Role>> is an example that represents the above mentioned material relationship. Textually, we have the following: *Workflow_Run* (X, Y) is true if and only if there is an *Executor* X and a *Concrete_Workflow* Y.

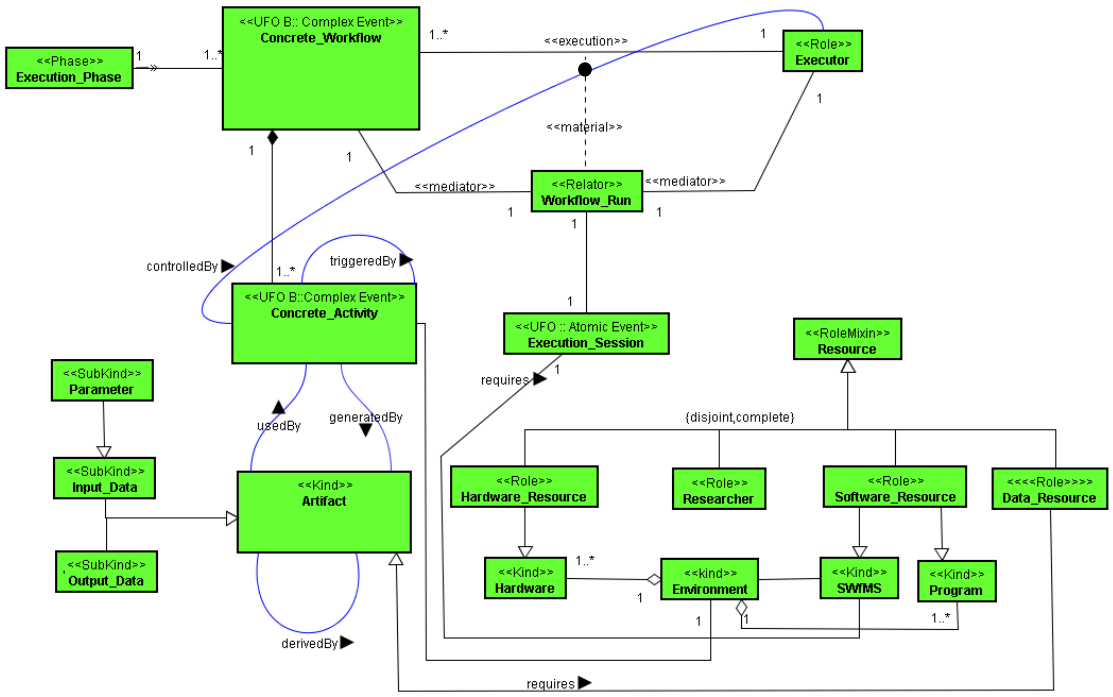


Figure 3. Fragment of the experiment execution sub-ontology

5. Related Work

There are few research initiatives committed with ontology-based models and formalizations of the *in silico* scientific experiments. A number of provenance ontologies have emerged from scratch or through conversion of existing provenance vocabularies. The main features with these provenance ontologies are: (i) they are focused on the provenance of digital objects (especially data) not the experiment; (ii) they leverage only the Web as the underlying infrastructure for data generation and data access; (iii) they encode computational semantics for provenance vocabularies using declarative representations; and finally (iv) they do not take into account a formal method like the one based on foundational ontology principles to validate its development.

The Open Provenance Model Ontology (Zhao, 2010) focuses solely on retrospective provenance of concrete scientific workflows. It defines a small set of core concepts for general entities (*artifacts*, *agents*, and *processes*) and relations in workflows, *e.g.*, "artifact wasGeneratedBy process" and "process wasControledBy agent". With a small number of concepts to represent such a complex domain, it results in concepts being semantically overloaded. It was developed by a community of workflow researchers and has multiple serialization profiles, including XML Schema and OWL. Currently, the OWL profile is still evolving to adapt the OPM specification to be a W3C standard.

The Proof Markup Language (PML) (McGuinness *et al.*, 2007) focuses on information manipulation processes such as logical reasoning, information extraction, and more recently, machine learning. It is modularized into several loosely coupled modules to: (i) annotate provenance metadata for sources of knowledge, (ii) encode information manipulation processes and data dependencies for deriving the conclusions or executing workflows and (iii) annotate trustworthiness assertions about knowledge and sources. PML uses a proof theoretic foundation for its model, its specification is synchronized with its own OWL ontology.

The Workflow Driven Ontology (Salayandia *et al.*, 2006) uses PML to represent abstract workflows, which is a plan for a workflow but not the execution of a workflow. It uses the *Method* concept to represent the class of actions to be executed and uses the *Data* concept to represent the class of data to be operated on by an action in the workflow. The Provenance Vocabulary (Hartig, Zhao, 2010) focuses on information manipulation. It consists of three modules: the core module defines basic concepts for representing data creation and data access processes, and the other two modules extend the core module by (i) adding classifications specific to Web information transfer and (ii) supporting authentication of information. It should be noted that this ontology uses OWL 2 language features. Provenir (Sahoo, Seth, 2009) is an ontology based on information manipulation. It reuses and redefines some provenance relations from the OBO Relation Ontology (Smith *et al.*, 2007), which defines generic binary relations without domain/range specifications.

Provenir, like the previously mentioned works, (re)defines some of the universal OPM classes (*process*, *artifacts* and *agent*) to its own purposes. These ontologies do not consider the role and the granularity of the different kinds of provenance metadata generated during the complete lifecycle of *in silico* experiments. However, as discussed, they tend to be too much driven by both requirements of specific applications and less committed to maximizing expressivity, clarity and truthfulness with regard to the domain of generic *in silico* scientific experiments. Furthermore, as a rule, such initiatives are not committed to the use of top-level ontologies such as UFO and, consequently, do not take advantage of neither its ontological foundation nor its subjacent ontological engineering approach to create clear conceptual models of high expressivity.

6. Conclusion

This article has presented a novel provenance ontology name Open Provenance Ontology. We advocate that the theory behind the OvO provides: (i) a knowledge repository about the different kinds of provenance of *in silico* scientific experiments and (ii) a reference conceptual model that may be used for promoting interoperability between distinct provenance systems.

Our contributions cover a gap in literature with regard to ontological approaches for modeling provenance metadata of scientific experiments. However, there is a need for future research on suitable languages to model phenomena such as the dynamics of the composition and execution of workflows in distributed environments. At this time, OvO is part of a distributed provenance gathering system named Matriohska, its modeling is being reviewed and part of it, implemented as OWL, and is under evaluation with *in silico* bioinformatics experiments at Fiocruz. Further details can be found at Cruz (2001). Additional future work includes: (i) the design and implementation of the OvO ontology in one or more codification languages (*e.g.*, F-Logic) as a proof of concept and (ii) the extension of the ontology for covering as far as possible novel situations.

References

Buneman, P., Khanna, S., W-C Tan, (2001) “Why and Where: A Characterization of Data Provenance”. In LNCS v. 1973 n. 2001 p. 316-330.

Cruz, S. M. S., Campos, M. L. M., Mattoso, M. (2009) “Towards a Taxonomy of Provenance in Scientific Workflow Management Systems”. In: SERVICES '09, p. 259-266.

Cruz, S.M.S., (2011) “Uma Estratégia De Apoio À Gerência De Dados De Proveniência Em Experimentos Científicos” Tese de Doutorado, PESC/COPPE-UFRJ , 234p.

Freire, J., Koop, D., Santos, E., Silva, C. T. (2008) "Provenance for Computational Tasks: A Survey," *Computing in Science and Engineering*, v. 10, n. 3, p. 11-21.

Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M. (2004) “Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. 1 ed. Springer, 2004.

Guizzardi, G. Wagner, G. (2005) “Some applications of a unified foundational ontology in business modeling”. Business Systems Analysis with Ontologies, chapter 13, p. 345–367.

Guizzardi, G. (2005)"Ontological Foundations for Structural Conceptual Models" CTIT PhD.-thesis series, No. 05-74, 441p.

Guizzardi, G. (2007). “On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models”. In Frontiers in Artifical Intelligence and Applications, Databases and Information Systems IV, IOS Press, Amsterdam.

Hartig, O., Zhao, J., Provenance Vocabulary Core Ontology Specification. (2010). Available at: <http://trdf.sourceforge.net/provenance/ns.html>.

Hey, T., Tansley, S., Tolle, K., (2009) “The Fourth Paradigm: Data-Intensive Scientific Discovery”. 1. ed. Redmond, Microsoft Research.

Jarrard, R. D. Scientific Methods, an online book. 1. ed. Dept. of Geology and Geophysics, University of Utah, 2001.

Mcguinness, D., Ding, L., Silva, P. P. (2007) “PML 2: A Modular Explanation Interlingua”. In: Proc of the 2007 Workshop on Explanation-aware Computing, pp. 49-55.

Mattoso, M., *et al.*. (2010) "Towards Supporting the Life Cycle of Large Scale Scientific Experiments". International Journal of Business Process Integration and Management, v. 5, p. 79-92.

Moreau, L., *et al.*., (2011) "The Open Provenance Model core specification (v1.1)". *Future Generation Computer Systems*, v. 27, n. 6, pp. 743-756.

Oinn, T. M., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, R.M., Carver, T., Glover, K., Pocock, M. R., Wipat, A. P. Li. (2004) “Taverna: a tool for the composition and enactment of bioinformatics workflows”. *Bioinformatics*, 20(17):3045– 3054.

Sahoo, S. S., Sheth, A., Corey, H. (2008) “Semantic Provenance for eScience: Managing the Deluge of Scientific Data” *Internet Computing*, IEEE, v. 12 n. 4, p. 46-54.

Sahoo, S. S., Shet, A., (2009) "Provenir ontology: Towards a Framework for eScience Provenance Management". In: Microsoft eScience Workshop.

Salayandia, L., da Silva, P.P. Gates, A. Q., Salcedo, G. F. (2006) “Workflow-Driven Ontologies: An Earth Sciences Case Study”. In E-SCIENCE '06 Proc. of the 2nd IEEE International Conference on e-Science and Grid Computing.

Stevens, R., Zhao, J., Goble, C, (2007) “Using provenance to manage knowledge of in silico experiments”, Brief Bioinformatics, v.8. n. 3, p. 183-194.

Smith, B., *et al.* (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology 25, 1251–1255.

Zhao, J. Open Provenance Model Vocabulary Specification. (2010). Available at <http://open-biomed.sourceforge.net/opmv/ns.html>.

Integrating Ecological Data Using Linked Data Principles

Ana Maria de C. Moura[1], Fabio Porto[1], Maira Poltosi[1] , Daniele C. Palazzi[1], Régis P. Magalhães[2], Vania Vidal[2]

[1]Extreme Data Lab (DEXL Lab)
National Laboratory of Scientific Computing (LNCC)
Petrópolis – RJ – Brazil
[2]Department of Computing
Federal University of Ceará (UFC)
{anamoura, fporto, maira, dpalazzi}@lncc.br, {regispires, vvidal}@lia.ufc.br

Abstract.*This paper presents a framework to manage, treat and integrate ecological data in the context of the PELD project, currently in development in Brazil. These data, which are produced and collected from different resources, are stored in distinct relational databases and transformed later into RDF triples, using a traditional relational-RDF mapping. Taxonomical, spatial and trophic relations are explored by means of ontological properties, which make it possible to discover interesting information about existing marine species of different bays in the country, illustrated by SPARQL queries. Additionally, the endpoint thus generated allows data to be accessed on the Web of data, as linked data.*

1. Introduction

Extensive information on policies, action programs, and environmental challenges in areas such as sustainable development, climate change, environmental law, and biodiversity, has become a great concern throughout the world. Different governmental agencies[1] and commissions[2,3] have been created for the purpose of defining strategies to preserve natural environment. Among these many policies, there is a strong concern on developing systems to organize and catalogue information about the existing natural reserves, such as minerals and biological ones, which involve fauna, flora and hydro resources, enabling a more accurate control of this information.

In Brazil, a great effort is being deployed in this direction through an important national project named PELD/Brazil[4] (Brazilian Long-Term Ecological Research Program). One of its main goals is to leverage ecological knowledge, so that important data can be provided to help, reinforce government decisions, and support research related to the management of natural resources, as well as to share this information among different sectors of society. PELD project currently counts on 29 collect sites, which are distributed along different Brazilian biomes, for the purpose of consolidating the existing knowledge about their composition and learning about ecosystems functioning. Having an integrated view of these ecological data sources and making

[1]http://www.environment-agency.gov.uk/
[2]http://ec.europa.eu/environment/index_en.htm
[3]http://www.princetontwp.org/environmain.html
[4]http://ppbio.inpa.gov.br/Port/projetosassociados/peld/

them available on the Web of data as a data set [Heath, Bizer 2011], would permit other ecologist researchers throughout the world to access, as well as to reference it to other data sets, dealing with similar subjects.

A PELD site can be considered as an integration of many sub-projects concerning distinct ecological issues. Since most of these PELD sites throughout the country are not still consolidated, or are in an initial development phase, in this paper we focus on the Guanabara PELD[5]. This PELD site aims at extending knowledge about the Guanabara Bay ecosystem and providing support for managing, structuring and publishing ecological data, as well as to be a source of answers to the anthropic and climatic impacts on the bay ecosystem. Currently a database project is being developed to manage, organize and access information about Guanabara Bay ecological data. However, since Guanabara PELD is developed by a large group of biologists, responsible by distinct domains (hydrology, planktons, fishes, ecology, etc.), data are produced independently, in different formats, and according to specific methodologies. Integrating and publishing all the data produced by these groups is crucial not only to provide a homogeneous view of this data, but also to make it available for other groups working in other PELDs throughout the country. This situation offers an interesting panorama to evaluate how efficient queries and reasoning will be in the face of a query federation pattern, where data are integrated according to the Linked Data (LD) strategy.

The main contribution of this work in comparison to other existing ecological information management systems (Ecoflora[6] [Cavalcanti 2005]) is: to integrate different ecological resources and to make them available on the Web of data, using LD principles; to provide reasoning capacity, i.e., to infer new information from the stored data. By providing an ontological representation of the data model, new relationships and instances may be inferred, taking into account transitive properties and hierarchies over the model concepts, allowing researchers to discover interesting data, such as, for example, information about specie's predators in different levels of a hierarchy.

In this paper we extend the integration framework [Vidal et al. 2011] and use some techniques to create the application ontologies. Query results are extracted from PELD data sources, integrated by QEF[7] framework [Porto et al 2007], and then visualized by the user as linked data.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 presents the framework architecture designed to integrate PELD resources. Section 4 describes some PELD application scenarios that will be used as study case for integration. Section 5 describes the scenario ontologies generated at each level of the proposed architecture, as well as the mappings rules between the domain and application ontologies. Section 6 shows how to answer user´s queries in this architecture as linked data, by presenting a query example over different PELD resources, executed in SPARQL. Finally, Section 7 concludes the paper with suggestions for future work.

---

[5] http://www.lncc.br/peldguanabara/index.php

[6] http://www.ecoflora.co.uk/

[7] http://146.134.234.248/QEF/index.html

2. Related Work

Several works such as Ecoflora[8], NRCS[8], AEZ[9], [Cavalcanti 2005], [Campos et al. 2009], [Manzi 2009] have been proposed to manage and share ecological data. However, they do not perform data integration on LD using multiple data sources. They also do not address inference provided by the use of ontologies or semantic web approaches. This paper intends to fill this gap, by proposing an integration approach based on LD, which enables ecological data to be analyzed, inferred and queried from different PELD sources. Below we present related works that address data integration over LD.

There are two possible approaches for data integration: materialized and virtual. The first approach collects, stores and accesses data in a central database. The main disadvantage of this approach is the replication of data, which in addition requires additional storage space and does not ensure the use of updated data in relation to the original datasources. LDIF [Schultz et al. 2011] is a framework that provides data integration through the use of the materialized approach. On the other hand, the virtual approach enables the execution of federated queries over a fixed set of datasources. Our work uses both the materialized and the virtual data integration approach. Jena ARQ[10] SPARQL, DARQ [Quilitz and Leser, 2008], SemWIQ [Langegger, 2010] and FedX [Schwarte 2011] are examples of systems that provide transparent access to RDF data sources, whose data can be retrieved using SPARQL. While some of them, such as SemWIQ, allows RDF schema or OWL ontologies to be used to describe the datasources, FedX transforms the original query into a federated query over the source ontologies. However, none of these tools can execute queries over a domain ontology with mappings for specific application ontologies.

The integration of scientific data in the context of linked data using the virtual approach has been discussed in [Gray et al. 2008]. In that paper the authors discuss the integration of astronomic databases using RDF as a common schema language and SPARQL as a query language. The authors adopt a peer-to-peer integration strategy, avoiding a global view agreement. In the proposed view, each database is exposed in RDF and alignment mappings define associations between databases.

The integration tools presented in this section require the manual definition of the datasources used in each query. It is also necessary to rewrite queries when a datasource schema changes. However, the generation of federated query plans from queries over a Domain Ontology can accomplish the semantic integration in virtual and automatic way. Queries over the domain ontology are also simpler and more stable than if they were made directly over the application ontologies.

3. Integration Architecture

The need to produce data in PELD projects in a homogenous format is a fundamental requirement when considering the generation of an integrated view of Brazilian ecosystems. In this context, RDF (Resource Description Framework) [Manolla, Miller 2004] has been used as a powerful strategy to interoperate, reason and publish data, besides enabling these data to connect with

[8] http://plants.usda.gov/java/

[9] http://www.fao.org/nr/land/databasesinformation-systems/aez-agro-ecological-zoning-system/en/

[10] http://jena.apache.org/documentation/query/

other resources of similar domains. Additionally, it enables the exploration and association among data, making use of SPARQL [Prud´hommeaux, Seaborne 2008].

Nevertheless, although the great benefits of RDF, there is a great concern when using it to deal with large volumes of data, since it may degrade performance [Gray et al. 2009]. This is why a current adopted strategy is to store data in relational databases. Moreover, publishing data according to the Linked Data best practices [Heath, Bizer 2011] solves part of the integration problem, which is to make data available in a common format. Ontologies come as a rescue ground from which integration becomes possible. They provide a common vocabulary to be shared among the different data sources. Thus, one needs to combine the publication of source data according to the LD best practices using RDF with a common shared vocabulary expressed as a domain ontology.

Figure 1 presents a three-level architecture used to integrate relational schemas as LD. It is based on mappings according to a mediated approach, and it has been extended from [Vidal et al 2011] to integrate the PELD databases as described below.

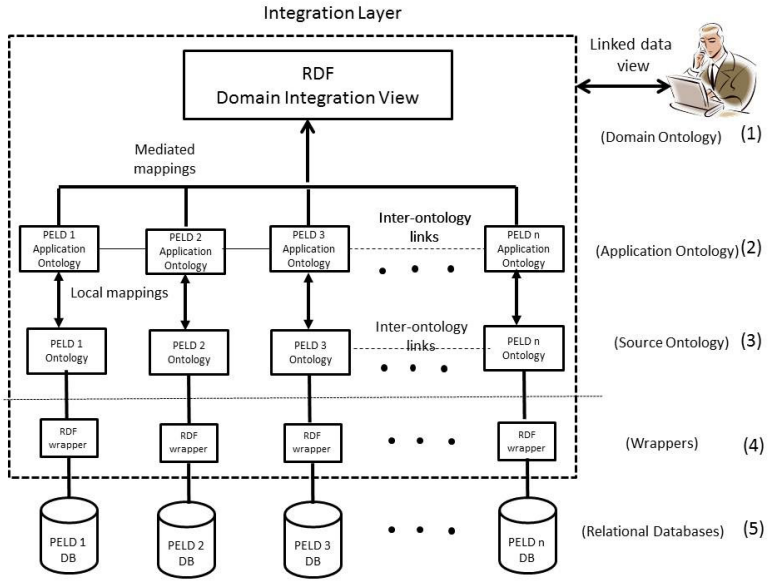


Figure 1. Three-level architecture for Linked Data Integration

The RDF Domain Integration View (1) is the Domain Ontology (DO) that represents the mediated schema. Designed by an expert user, it provides a conceptual representation of a specific domain, which comprises a global shared vocabulary and constraints. Each PELD relational database (5) is transformed into RDF by a specific wrapper (4) (see section 4.4) and becomes a source ontology (3), which is then rewritten as a PELD application ontology (APO) (2). It is worth observing that each APO describes a source ontology according to the principles of LD, which is a subset of the DO ontology. Application ontologies help breaking the query answering problem in two steps: (i) a query is submitted to the mediated schema, i.e., to the domain integration view, and by using mediated mappings, the query over the integration view is rewritten in terms of the application ontologies. As an example, consider queries over the Sample concept (Figure 2), which are rewritten as unions of AO; then (ii) based on the rewritten query an execution plan is generated, in which references between APOs become joins, and each sub-query, completely covered by an AO is rewritten using local mappings,

and then submitted to the corresponding PELD local databases to retrieve information and deliver an integrated query answer to the user as LD. This step by step procedure is better described in section 6.

4. Application Scenarios

This section describes the Guanabara PELD scenarios that will be used for integration, based on the architecture depicted in Figure 1.

Guanabara PELD aims at getting biotic data from samples extracted from the bay water and from fishing resources. The living organisms are hierarchically classified in a taxonomy. The first level corresponds to the *Kingdom*, which is decomposed into *Phylums* and successively into *classes*, *orders, families*, *genders* and *species*. Each level has respectively its own subdivisions. Any level within this classification is called a *taxon*. There exist differences in the levels concerning each organism. Some of them have been reclassified, and in this case, both classifications are kept, and a synonymous relation is established between them.

In the context of ecological data analysis, some important features deserve some attention. Geographical region information identifies a target ecosystem and is used for selecting and classifying events according to their location. On the other hand, trophic relations are fundamental for the ecosystem study. Finally, the taxonomy enables a hierarchical analysis of the species. The analysis of these aspects may be explored by the use of inference in an integrated way.

The main characteristics of each scenario are described next.

- *Plankton:* in the plankton scenario, a sample data takes into account temporal (data and time) and spatial (latitude, longitude and profundity) information, as well as methods used for sample collect and conservation, atmospheric, and maritime conditions during each collect. For each analysis performed, data, sample and the applied method are registered. Biomass measurements of organisms found in the samples can be done at specie level or at the taxonomy highest level;
- *Community Fish:* besides temporal and spatial information, this application scenario stores the *fishing method* used to catch fishes, taking into account two different depths (initial and final). It is worth observing that collected fishes are divided into three samples, from which the total weight and number of individuals are analyzed for each *taxon* found in the collect process;
- *Catfish Genidens:* differently from the previous scenarios, this application scenario analyzes each specific specie individually, considering not only spatial and temporal references, but also the *fishing method* employed in the collect process, the specie *weight, length* and *gender*.

5. Domain and Application Ontologies

Based on the application scenarios described above, this section describes the ontologies generated at each level of the framework architecture presented in Figure 1.

Domain Ontology(DO)

Since in this paper the main purpose is not ontology design, we assume the domain ontology is provided by the user. Figure 2 presents the conceptual representation of the PELD domain ontology, referenced in our architecture as RDF Domain Integration

View. The namespace prefix "d" is used to refer to the vocabulary of this domain ontology. Since most of the class properties are self-described, we just give a few examples of the class properties. Thus, *d:collect_method* is defined as a *datatype property* with domain *d:Sample* and range *string; d:has_predator* is an object type property with domain *d:Trophic_Chain* and range *d:Taxon;* and *d:has_pl_analysis* is also defined as an object property, with domain *d:Plankton_sample* and range *d:Pl_analysis.*

Application Ontology(AO)

As mentioned in section 1, PELD sites are composed of different PELD subprojects. Each such PELD subproject takes part in the PELD data integration, by providing their local data published in RDF, which is rewritten as an AO, using a subset vocabulary of the DO. As in a federated database, an application ontology may be seen as an external ontology that takes part in the integrated schema, i.e., the domain ontology. Figure 3 presents a conceptual representation of the PELD AOs associated with the application scenarios described above comprising five ontologies: *Plankton, Catfish Genidens*, *Community Fishes, Region and Taxon*, each one having the following namespace prefixes: "apl:", "acf:","aco:", "r", and "tx" respectively. As mentioned before, the vocabulary of an application ontology consists of classes and properties that are subset of the domain ontology. Thus, access to the local data is done through direct mappings and the integration work becomes facilitated.

Based on the work proposed in [Vidal et al 2011], Figure 4 presents the list of the rules defined for the mapping between the APO and the DO. Due to space restriction we present only the mapping rules of *Plankton* ontology and we refer the reader to the above reference for more details on the definition of these rules, which is not in the scope of this paper.

It is worth mentioning that since the ontologies Region and Taxon represent data that are not frequently changed, they are previously materialized and stored locally as RDF triples in a repository, also as AOs. Thus, they are accessed whenever required and joined together with the other APOs that are virtually retrieved, as described in section 6.2.

6. Querying over the Framework Architecture

The main purpose of the proposed integration framework architecture is to answer user´s queries in terms of a domain ontology. Through the unified view exposed by the DO, researchers can access PELD subproject data transparently independently of local particularities. In order to deliver data, the data integration framework must be supported by a data integration engine that processes user´s query requests and returns results dealing with necessary data translations and access to source data[11]. In the context of this paper, ontologies in all architecture levels are homogeneously expressed in RDF. Thus, user requests may be submitted to the data integration system using SPARQL. The query expression is transformed into sub-queries over the application ontologies exposed as RDF triples by the D2RQ [Bizer et al. 2006] engine from the source databases.

The QEF system developed at DEXL laboratory has been used as the data integration engine. QEF is an extensible query engine that supports user-defined

[11] In the current version QEF does not rewrite queries yet. This is considered as a future work.

algebras and data structures. In order to support PELD data integration, a new version named QEF-LD [Magalhães 2012] has extended QEF. This new version includes linked data algebraic operators, and wrappers that submit AO sub-queries to a D2R endpoint. The latter exports local databases as virtual AOs.

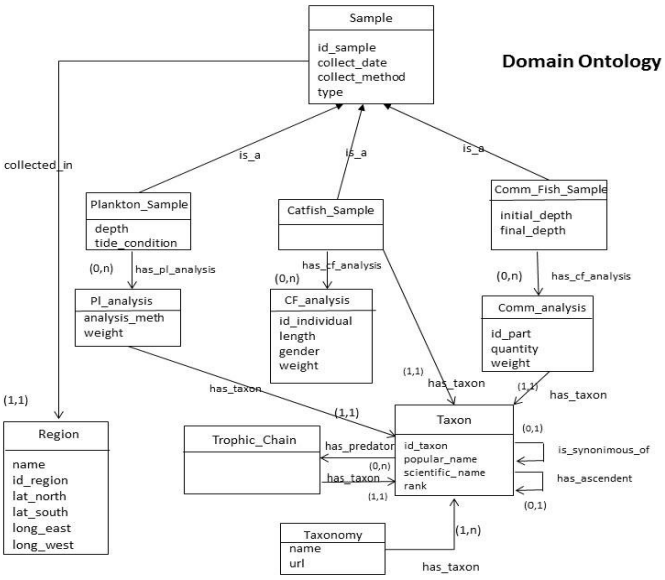


Figure 2. PELD domain ontology

In scenarios where a domain ontology query is translated into sub-queries over more than one application ontology, results are combined by the Union operator and returned to the user in a single result set.

Considering the strategy developed in [Vidal et al 2011], the following algorithm is performed:

- The user submits a SPARQL query to the data integration system expressed in terms of a domain ontology. Then, according to the mediated mappings, an integrated query execution plan is generated according to the following steps:
  a. References to the concepts Region and Taxonomy in the query, which are shared by the AOs, are mapped to BindJoins [Magalhães 2012] between the source AO and the shared databases (i.e. Region or Taxonomy).
  b. Each sub-query is submitted to a data source. D2R endpoints translate the submitted queries to the corresponding local database queries. The Region and Taxonomy AOs are materialized as RDF sources and joined with AO ontology concepts through SPARQL queries.
  c. Once the results are obtained, QEF applies the joins and unions handling in the final result. A query over the Sample concept is rewritten as Unions of subqueries over each APOs (see figure 4 (a)), according to the mappings presented in Figure 4(b), respectively. This step is not currently supported by QEF-LD [Magalhães 2012].

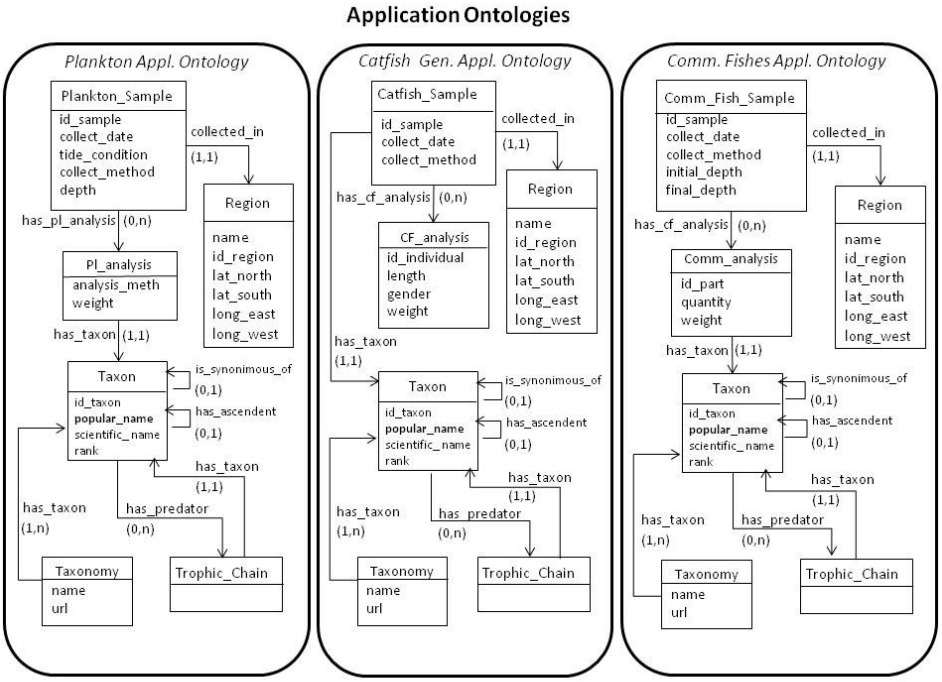


Figure 3. PELD application ontologies

d:Sample(p) ⇐ apl: Plankton_Sample(pls) U acf: Catfish_Sample(c) U
aco: Comm_Fish_Sample(co)

Figure 4(a). Sample DO expressed as the union of the different sample species of APO ontologies

1. d:Pl_analysis(pl) ⇐ apl:Pl_analysis(pl)
2. d: Plankton_Sample(pls) ⇐ apl: Plankton_Sample(pls)
3. d:id_sample(p,id) ⇐ apl: id_sample(pls,id), apl: Plankton_Sample(pls)
4. d:col_date(p,dt) ⇐ apl: collect_date(pls,dt), apl:Plankton_Sample(pls)
5. d:collect_method (p,cm) ⇐ apl: collect_method(pls,cm), apl: Plankton_Sample(pls)
6. d:depth (pls,d) ⇐ apl:depth(pls,d), apl: Plankton_Sample(pls)
7. d:tide_condition (pls,tc) ⇐ apl: tide_condition(pls,tc), apl: Plankton_Sample(pls)
8. d:collect_in (p,l) ⇐ apl: collect_in(pls,l), apl:Plankton_Sample(pls), Region(l)
9. d:analysis_meth (pl,am) ⇐ apl: analysis_meth(pl,am), apl:Pl_analysis(pl)
10. d:weight (pl,w) ⇐ apl: weight(pl,w), apl:Pl_analysis(pl)
11. d:has_taxon (pl,tx) ⇐ apl:has_taxon(pl,tx) , apl:Plankton_analysis(pl), Taxon(tx)
12. d:has_pl_analysis (p,pl) ⇐ apl:has_pl_analysis(pls,pl)
13. d:type (p,´plankton`) ⇐ apl: Plankton_Sample(p)

Figure 4(b). Mapping rules from the Plankton APO to DO

6.1 Submitting a Query

According to the strategy presented above, the following query has been submitted to the proposed framework: "*Get the species found at Paquetá Island in 2004, their synonyms and predators*". In the following paragraphs the transformation process for answering this query is described, step by step.

i) The main query (Q) is expressed in terms of the domain ontology, which comprises the union of the 3 species: Planktons, Catfish and Comm. Fish.

```
Select distinct ?name ?name_syn
?name_pred
Where {
{
  ?s d:collected_in ?r .
  ?s d:collect_date ?dt.
  ?r d:name ?reg.
  ?p d:is_a ?s .
  ?p d:has_pl_analysis ?pl .
  ?pl d:id_taxon ?tx.
  ?tx d:popular_name ?name
}
Union {
  ?p  d:is_a ?s .
  ?cf  d:id_taxon ?tx .
  ?tx d:popular_name ?name .
}
Union {
  ?p d:is_a ?s .
  ?p d:has_cf_analysis ?pl .
  ?pl d:id_taxon ?tx .
  ?tx d:popular_name ?name .
}
Optional {
  ?tx d:has_predator ?pred .
  ?pred d:has_taxon ?idpred .
  ?idpred d:popular_name ?name_pred .
}
Optional {
  ?syn d:is_synonimous-of  ?tx;
       d:popular_name ?name_syn .
}
Filter (?reg = "Paqueta" && ?dt = 2004 )
}
order by ?name
```

ii) Query Q is rewritten as the union of three subqueries Q_1, Q_2 and Q_3, which aim at extracting data from *Plankton, Catfish Genidens,* and *Comm. Fish* application ontologies, region and taxon, respectively (Figure 5).

6.2 Executing a Query in QEF

As mentioned before, part of the application ontologies are stored in RDF tuples as materialized views. Such characteristic requires an execution plan for each query Q_i (Figure 6(a)), in order to guide QEF into the correct execution of the algebra operators sequence over the local data sources.

In order to exemplify this step, consider query Q'_1 the Q_1 version about *Planktons* that will be submitted to QEF. Similarly to Q_2 and Q_3, these queries use both virtual and materialized information. In other to describe the step by step execution procedure performed by QEF, Figures 6 (a) and (b) present respectively a Q'_i query execution plan for each Q_i, and each corresponding SPARQL query. Figures 7, 8, and 9 present, respectively, the results of $Q'1$, $Q'2$ and $Q'3$.

Final results (Figure 10) are obtained from $Q'1 \cup Q'2 \cup Q'3$, having duplicated values discarded.

7. Conclusion

This paper reports on the application of the aforementioned data integration framework to the ecological domain and the extension of QEF, a data integration system, to answer queries on heterogeneous ecological databases using this framework. A complete data integration scenario is discussed based on the challenges involved in publishing ecological data produced by the PELD Guanabara project, in Brazil.

Based on the data integration framework, a set of PELD subproject databases stored in relational databases are transformed into RDF as endpoints via D2RQ, which enable an integrated view over the data resources via SPARQL queries. The results indicate that the proposed data integration framework is promising and that shall be adopted as a standard for more complex ecological database integration scenarios.

Q_1	Q_2	Q_3

Select distinct ?name ?name_syn ?name_pred
Where {
?s apl:collected_in ?r.
?s apl:col_date ?dt.
?r r:name?reg.
?s apl:has_pl_analysis ?a.
?a tx:has_taxon ?tx.
?tx tx:popular_name ?name.
Optional {
?tx tx:has_predator ?pred.
?pred tx:has_taxon ?idpred.
?idpred tx:popular_name ?name_pred.
}
Optional {
?syn tx:is_synonimous_of ?tx;
tx:popular_name ?name_syn}.
Filter (?reg = "Paqueta" && ?dt = 2004).
} order by ?name
	Select distinct ?name ?name_syn ?name_pred
	Where {
	?s acf:collected_in?r.
	?s acf:col_date ?dt.
	?r r:name ?reg.
	?s tx:has_taxon ?tx.
	?tx tx:popular_name ?name.
	Optional {
	?tx tx:has_predator ?pred.
	?pred tx:has_taxon ?idpred .
	?idpred tx:popular_name ?name_pred.
	}
	Optional {
	?syn tx:is_synonimous_of ?tx;
	tx:popular_name ?name_syn
	}
	Filter (?reg = "Paqueta" && ?dt = 2004).
	}
	order by ?name
		Select distinct ?name ?name_syn ?name_pred
		Where {
		?s aco:collected_in ?r.
		?s aco:col_date ?dt.
		?r aco:name?reg.
		?s aco:has_cf_analysis ?a.
		?tx tx:popular_name ?name.
		?a tx:has_taxon ?tx.
		Optional {
		?tx tx:has_predator ?pred.
		?pred tx:has_taxon ?idpred.
		?idpred tx:popular_name ?name_pred.
		}
		Optional {
		?syn tx:is_synonimous_of ?tx;
		tx:popular_name ?name_syn
		}
		Filter (?reg = "Paqueta" && ?dt = 2004).
		}
		order by ?name

Figure 5. Q_i SPARQL query

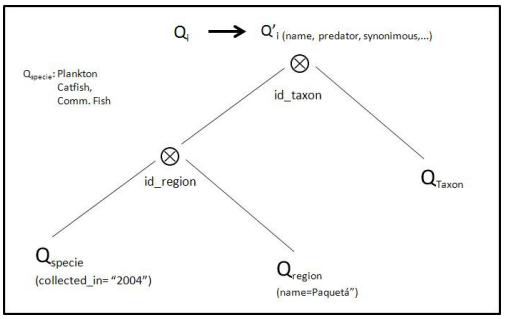


Figure 6(a). Q'_i execution plan

Q'_1 (Planktons)	Q'_2 (Catfish)	Q'_3 (Comm.fish)

$Q_{plankton}$: Select ?id_taxon, ?id_region
Where {
?s apl:collected_date ?dt.
?s apl:collected_in ?id_region.
?s apl:has_pl_analysis ?id_an.
?id_an tx:has_taxon ?id_taxon.
Filter (?dt =2004).
}
--- 	
Q_{region}: Select ?id_region
Where {
	$Q_{catfish}$: Select ?id_taxon, ?id_region
	Where {
	?s acf:collected_date ?dt.
	?s acf:collected_in ?id_region.
	?s tx:has_taxon ?id_taxon.
	Filter (?dt =2004).
	}
	--- 	
	Q_{region}: Select ?id_region
	Where {
		$Q_{commfish}$: Select ?id_taxon, ?id_region
		Where {
		?s aco:collected_date ?dt.
		?s aco:collected_in ?id_region.
		?s aco:has_cf_analysis ?id_an.
		?id_an tx:has_taxon ?id_taxon.
		Filter (?dt =2004).
		}
		--- 	
		Q_{region}: Select ?id_region
		Where {

?id_region r:name ?n.
?r r:id_region ?id_region.
Filter (?n, “Paquetá”).}
---------- 	
Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
Where {
?x tx:id_taxon ?id_taxon.
?id_taxon tx:scientific_name ?name.
?id_taxon tx:has_predator ?pred.
?pred tx:has_taxon ?tx_pred.
?tx_pred tx:scientific_name ?name_pred.
?id_taxon tx:is_synonimous_of ?syn_tax.
?syn_tx tx:scientific_name ?name_syn.
}
	?id_region r:name ?n
	?r r:id_region ?id_region.
	Filter (?n, “Paquetá”).}
	---------- 	
	Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
	Where {
	?x tx:id_taxon ?id_taxon.
	?id_taxon tx:scientific_name ?name.
	?id_taxon tx:has_predator ?pred.
	?pred tx:has_taxon ?tx_pred.
	?tx_pred tx:scientific_name ?name_pred.
	?id_taxon tx:is_synonimous_of ?syn_tax.
	?syn_tx tx:scientific_name ?name_syn.
	}
		?id_region r:name ?n
		?r r:id_region ?id_region.
		Filter (?n, “Paquetá”).}
		---------- 	
		Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
		Where {
		?x tx:id_taxon ?id_taxon.
		?id_taxon tx:scientific_name ?name.
		?id_taxon tx:has_predator ?pred.
		?pred tx:has_taxon ?tx_pred.
		?tx_pred tx:scientific_name ?name_pred.
		?id_taxon tx:is_synonimous_of ?syn_tax.
		?syn_tx tx:scientific_name ?name_syn.
		}

Figure 6(b). Q´i SPARQL query

SPARQL results:

name	name_syn	name_pred
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	"Argentine menhaden"
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	-
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	"Marinis anchovy"
"Oithona plumifera"	-	-
"Oithona plumifera"	-	"Argentine menhaden"
"Oithona plumifera"	-	"Marinis anchovy"
"Oitona hebes"	-	-
"Oitona hebes"	-	"Argentine menhaden"
"Oitona hebes"	-	"Marinis anchovy"
"Temora turbinata"	-	-
"Temora turbinata"	-	"Argentine menhaden"
"Temora turbinata"	-	"Marinis anchovy"

Figure 7. Results of Q´1

SPARQL results:

name	name_syn	name_pred
"Argentine menhaden"	-	-
"Argentine menhaden"	-	"Marine catfish"
"Argentine menhaden"	-	"Whitemouth croaker"
"Franciscana"	-	-
"Marine catfish"	-	-
"Marinis anchovy"	-	-
"Marinis anchovy"	-	"Marine catfish"
"Marinis anchovy"	-	"Whitemouth croaker"
"Marinis anchovy"	-	"Franciscana (Stenodelphis)"
"Whitemouth croaker"	-	-
"Whitemouth croaker"	-	"Franciscana (Stenodelphis)"

Figure 9. Results of Q´3

SPARQL results:

name	name_syn	name_pred
"Marine catfish"	-	-

Figure 8. Results of Q´2

Num	Name	Synonym	Predator Name
1	Marine catfish	null	null
2	Acartia tonsa	Acartia (Acanthacartia) tonsa	Argentine menhaden
3	Acartia tonsa	Acartia (Acanthacartia) tonsa	null
4	Acartia tonsa	Acartia (Acanthacartia) tonsa	Marinis anchovy
5	Oithona plumifera	null	null
6	Oithona plumifera	null	Argentine menhaden
7	Oithona plumifera	null	Marinis anchovy
8	Oitona hebes	null	null
9	Oitona hebes	null	Argentine menhaden
10	Oitona hebes	null	Marinis anchovy
11	Temora turbinata	null	null
12	Temora turbinata	null	Argentine menhaden
13	Temora turbinata	null	Marinis anchovy
14	Argentine menhaden	null	null
15	Argentine menhaden	null	Marine catfish
16	Argentine menhaden	null	Whitemouth croaker
17	Franciscana	null	null
18	Marine catfish	null	null
19	Marinis anchovy	null	null
20	Marinis anchovy	null	Marine catfish
21	Marinis anchovy	null	Whitemouth croaker
22	Marinis anchovy	null	Franciscana (Stenodelphis)
23	Whitemouth croaker	null	null
24	Whitemouth croaker	null	Franciscana (Stenodelphis)

Figure 10. Final Results

ACKNOWLEDGEMENTS

This work has been partially supported by CNPq through its Institutional Capacity Program (Proc. 382.489/09-8) and Productivity Research fellowship (Proc. 309502/2009-8).

References

Bizer C., Health T., Berners-Lee T. D2R Server – Publishing relational databases on the Web as SPARQL endpoints. Proc. of the 15th International World Wide Web Conference, Edinburgh, Scotland, 2006.

Campos, S.R., Martinhago A.Z., Massahud R.T., França A.M., Prieto L. E., Mendes J.D.C. Database modeling of the economic ecological zoning of Minas Gerais using UML-GeoFrame (in Portuguese). Proc. of the XIV Brazilian Symposium of Remote Sensoring, Natal, Brasil, 25-30 April, 2009, INPE, p. 4943-4949.

Cavalcanti, M. J. Database on Amazon biodiversity: experience on the Biotupé project. Biotupé: Physical environment, biological diversity and sociocultural of Low Negro River, central Amazon (*in Portuguese*). Santos-Silva, Aprile, Scudeller, Editora INPA, Manaus, 2005.

Gray A. J. G., Gray N., Ounis I. Can RDB2RDF tools feasibily expose large science archives for data integration? The Semantic Web: Research and Applications – LNCS, Vol. 5554/2009, 491-505, 2009. DOI: 10.1007/978-3-642-02121-3_37.

Heath, T., Bizer C. Linked Data: evolving the Web into a global data space (1st edition). Synthesis lectures on the semantic Web: theory and technology, 1:1, 1-136. Morgan & Claypool ed., 2011.

Langegger, A., Wöß, W., Blöchl, M. 2008. A Semantic Web Middleware for Virtual Data Integration on the Web. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science. Springer Verlag, pp. 493–507.

Magalhães, R. P. Um Ambiente para Processamento de Consultas Federadas em Linked Data Mashups. M.S. thesis, Universidade Federal do Ceará, 2012.

Manola, F. and Miller, E. RDF primer. W3C Recommendation, February, 2004. Available at: http://www.w3.org/TR/rdf-primer.

Manzi, A. Data management of Brazilian long-term ecological research projects (*in Portuguese*), research project, Edital MCT/CNPq Nº 59/2009 – PELD support proposals, 2009.

Porto F., Tajmouati O., Silva V. F. V., Schulze B., Ayres F. V. M. QEF - supporting complex query applications, 7th IEEE International Symposium on Cluster Computing and the Grid — CCGrid 2007 , Rio de Janeiro, Brazil, pp. 846-851.

Prud´hommeaux, E. and Seaborne, A. 2008. Sparql Query Language for RDF. W3C Recommendation. Available at: http://www.w3.org/TR/rdf-sparql-query/.

Prud'hommeaux, E. And Buil-Aranda, C. SPARQL 1.1 Federated Query. http://www.w3.org/TR/sparql11-federated-query/, 2011.

Quilitz, B. and Leser, U. 2008. Querying Distributed RDF Data Sources with SPARQL. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science, Springer Verlag, pp. 524–538 (2008).

Schultz, A., Matteini, A., Isele, R., Bizer, C., and Becker, C. LDIF : Linked Data Integration Framework. In Proceedings of the 11th Interational Semantic Web Conference ISWC2011. pp. 1–4, 2011.

Schwarte, A., Haase, P., Hose, K., Schenkel, R., and Schmidt, M. Fedx: optimization techniques for federated query processing on linked data. In Proceedings of the 10th international conference on The semantic web - Volume Part I. ISWC'11. Springer-Verlag, Berlin, Heidelberg, pp. 601–616, 2011.

Vidal, V.M.P, Macêdo, J.A.F., Pinheiro, J. C., Casanova, M. A., Porto F. Query processing in a mediator based framework for linked data integration. IJBDCN 7(2): 29-47, 2011.

Part II
Short Papers

Extração automática de termos candidatos às ontologias: um estudo de caso no domínio da hemoterapia

Fabrício M. Mendonça[1], Maurício B. Almeida[1],
Renato R. Souza[2], Daniela L. Silva[1,3]

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

[2]Escola de Matemática Aplicada – Fundação Getúlio Vargas
Praia do Botafogo – CEP – Rio de Janeiro – Brasil

[3]Departamento de Biblioteconomia – Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514 - Goiabeiras – 29.075-910 – Vitória – Brasil

fabriciommendonca@gmail.com, mba@eci.ufmg.br, renato.souza@fgv.br, danielalucas@hotmail.com

Abstract. *This paper describes a case study conducted within the domain of blood transfusion aiming at non-exhaustively extraction of candidate terms for an ontology of human blood. The process involved both the construction of a corpus and its automatic processing, and the retrieval of specialized terms. As our main result, we have obtained candidate medical terms to be used in a ontology of blood transfusion processes.*

Resumo. *O presente artigo descreve um estudo de caso realizado no domínio de hemoterapia para a extração automática e não exaustiva de termos candidatos às ontologias sobre o sangue humano. O processo envolveu a construção de um corpus, seu processamento por ferramenta automática e a recuperação de termos médicos. Ao final do experimento, obtiveram-se os termos médicos candidatos a ontologia sobre processos hemoterapêuticos.*

1. Introdução

A ampla utilização da internet e das novas tecnologias de informação, tais como os dispositivos móveis, as redes sociais, os sistemas de informação eletrônicos, têm mudado a forma do ser humano manipular informação. Uma das consequências destas mudanças é a necessidade de novas abordagens para organização da informação.

No âmbito da medicina, a extração de informação a partir de textos tem sido abordada de duas formas principais [Friedman e Hripcsak 1999]: (i) a abordagem manual – por vezes denominada curadoria – realizada por especialistas capacitados em abstrair informações dos textos médicos e correspondê-los a conceitos de terminologias médicas; e (ii) a abordagem automática, que consiste na extração de termos médicos realizada, normalmente por ferramentas de processamento de linguagem natural (PLN).

Considerando as possíveis potencialidades da extração automática na organização e representação da informação médica [Winnenburg et al. 2008], o presente

artigo descreve um estudo de caso no domínio da hemoterapia, em que foi realizada a extração de termos médicos de um corpus para uso em ontologia sobre os processos da manipulação do sangue humano [Almeida et al. 2010]. Optou-se por utilizar ontologias para modelagem do conhecimento na área, devido às suas vantagens significativas em relação à modelagem conceitual tradicional, caracterizada por ser *ad hoc* e orientada por casos [Smith e Welty 2001].

A ferramenta *Sketch Engine*[1] foi utilizada para a construção e o processamento morfossintático de um corpus sobre o sangue humano, a partir de textos especializados. A extração de termos se baseou em uma lista de sufixos médicos relevantes no domínio, sugerida por especialistas com base em seus conhecimentos, além de referências da área.

O presente artigo está estruturado nas seguintes seções: na seção 2 discorre-se sobre o uso de ferramentas de PLN para extração de informação e construção de ontologias; na seção 3 apresenta-se a metodologia usada para a extração de termos do corpus como possíveis candidatos à ontologia; na seção 4 são descritos os resultados obtidos com o experimento realizado; e, na seção 5 são apresentadas conclusões acerca deste trabalho e os trabalhos futuros previstos nesta pesquisa.

2. Uso de ferramentas de PLN na extração e organização da informação

Embora existam ferramentas de PLN disponíveis para extração de informação e auxílio na construção de ontologias [Buitelaar et al. 2003], [Cimiano e Volker 2005], [Wätcher e Schroeder, 2010], os resultados nem sempre são satisfatórios para tratar a complexidade envolvida na aquisição de conhecimento para ontologias. Na maioria dos casos, torna-se necessária a intervenção humana para ajuste dos termos à ontologia [Smith et al. 2005]. Ainda assim, as ferramentas de PLN são consideradas úteis para diversas tarefas na construção de ontologias e no processo de curadoria [Buitelaar et al 2003] [Winnenburg et al 2008], principalmente na fase de aquisição de conhecimento.

Afora essa discussão, fundamental é ressaltar aspectos essenciais para a extração de informação de textos através das ferramentas de PLN, que se referem: à definição ou criação de um corpus no domínio sob estudo e aos procedimentos usados para a análise e anotação linguística do corpus (processamento). Um corpus é "um conjunto estruturado de grandes dimensões de textos, eletronicamente armazenados e processados, usado para um propósito definido e que seja representativo do domínio sob estudo" [McEnery e Wilson 2011]. Nesse sentido, nem todo conjunto de textos eletrônicos é propriamente um corpus. De fato, os critérios relevantes para a construção de corpora envolvem *autenticidade, tamanho, amostragem, representatividade* e *balanceamento* [Biber 1993] [Tognini-Bonelli 2001].

Outro aspecto fundamental refere-se ao trabalho de análise linguística do corpus, que assim como sua criação, pode ser feito manualmente ou com o uso de ferramentas automatizadas. Essa análise envolve imprescindivelmente a etapa de codificação do corpus, bem como a anotação ou etiquetagem dos seus elementos. Para o processo de anotação de corpora também existem princípios como a *recuperabilidade* e a *capacidade de extração* [Leech 1993].

[1] A ferramenta está disponível em: http://Sketch Engine.co.uk/. Acesso em 29 de Junho de 2012.

3. Metodologia para extração dos termos candidatos

O objetivo da presente seção é apresentar a metodologia empregada para extrair do corpus formado os termos candidatos à ontologia de processos do sangue humano. A abordagem utilizada é considerada semi-automática, devido à intervenção humana, e envolveu três etapas principais: (i) construção de um corpus no domínio do sangue, utilizando-se uma ferramenta automática (seção 3.1); (ii) processamento automático do corpus através de análise morfossintática (seção 3.2); e (iii) cálculo da frequência dos termos candidatos à ontologia (seção 3.3), por meio de tarefas manuais e automáticas.

3.1. Construção de um corpus no domínio do sangue

Os textos escolhidos para compor o corpus no domínio do sangue foram extraídos de um manual técnico sobre os padrões de qualidade para manipulação do sangue humano da instituição americana AABB[2] *Technical Manual*, 17ª edição. Nesse sentido, procura-se atender aos critérios citados para construção de corpus.

Com a amostra de textos selecionada partiu-se para a criação do *corpus* no *Sketch Engine*, que é capaz de processar textos em formato *pdf* para construção de corpora. Dos 32 capítulos da 17ª edição do *AABB's Technical Manual*, 27 foram incluídos na formação do corpus, totalizando 369.741 *tokens* identificados.

3.2. Análise morfossintática do *Blood Corpus*

A análise morfossintática do corpus foi realizada utilizando-se também a ferramenta *Sketch Engine*. Essa etapa permitiu a identificação e anotação linguística dos *tokens* do *Blood Corpus* (BC).

Para anotação do corpus, o *Sketch Engine* utiliza: (i) a linguagem de marcação XML na anotação das informações linguísticas dos elementos do texto; e (ii) princípios de anotação de corpora do padrão internacional *Text Encoding Initiative*[3] (TEI). Já o tipo de anotação, realizada pela ferramenta no corpus BC, corresponde à anotação *Part-of-Speech (POS) Tagger*, que inclui a etapa de lematização e a anotação das categorias morfo-sintáticas dos elementos do texto. Nessa anotação, cada item lexical é associado a apenas uma categoria gramatical (etiqueta) de acordo com seu uso na frase.

3.3. Cálculo da frequência dos termos candidatos a processos do sangue

Após a construção e a anotação do corpus, o passo seguinte foi extrair termos candidatos à ontologia dos processos sobre o sangue humano, conforme segue.

A estratégia semi-automática utilizada envolveu três passos: (i) seleção manual de sufixos médicos que identifiquem processos; (ii) cálculo automático da frequência dos termos que possuem tais sufixos no corpus; e (iii) agrupamento manual dos termos recuperados em classes semânticas de acordo com o sufixo que possuem.

---

[2] AABB é uma associação internacional que conta com 2000 instituições de saúde e 8000 profissionais vinculados, originários de 80 diferentes países do mundo [AABB 2012].

[3] O *Text Encoding Initiative* (TEI): envolve três importantes associações de linguística computacional do mundo, para a criação de formatos padronizados de anotação [McEnery e Wilson 2001].

A justificativa pela escolha dos sufixos dos termos como base para cálculo de frequência e, consequentemente, para extração do corpus, deve-se ao fato de que essa parte da palavra, normalmente, representa o significado (semântica) de um termo médico. Desta forma, o **sufixo** indica o procedimento, a condição ou a doença representada pelo termo. Em "*polycythaemia*", por exemplo, o prefixo *poly* indica *muitos,* a raiz *cythea* representa *célula* como parte do corpo humano onde o processo ocorre e o sufixo *emia* é relativo *à falta de algo*.

Nesse sentido, consultaram-se especialistas e referências na área para a criação de uma lista de sufixos que representem procedimentos médicos na área de hemoterapia. Os sufixos selecionados foram: *-apheresis, -centesis, -desis, -ectomy, -opsy, -rrhaphy, -metry, -scopy, -oscopy, -otomy, -ostomy, -pexy* e *-plasty*.

De posse dos sufixos médicos, procedeu-se com a construção de expressões regulares utilizando a linguagem *Corpus Query Language* (CQL) dentro da ferramenta *Sketch Engine*. A execução de tais expressões na ferramenta permitiu recuperar todos os termos do corpus que possuem esses sufixos, bem como a frequência absoluta do termo no corpus, ou seja, seu número de ocorrências.

O passo seguinte para a extração dos termos do corpus BC foi selecionar da lista de frequência calculada apenas aqueles termos com maior frequência e agrupá-los em sua classe semântica correspondente, de acordo com o significado do seu sufixo. A execução deste último passo possibilitou a sugestão de termos candidatos à ontologia.

4. Resultados parciais

Nesta seção, apresentam-se os resultados obtidos até o momento com a extração automática de termos do *corpus* e sua representação como classes de uma ontologia sobre os processos de manipulação do sangue humano.

O agrupamento manual dos termos recuperados do corpus BC em classes semânticas, de acordo com os sufixos que os compõem, é mostrado na tabela 1. Nela, os números entre parênteses indicam a frequência do termo no corpus. Considerou-se como frequência mínima para este agrupamento valores maiores ou iguais a três ocorrências, assim os demais termos recuperados foram descartados.

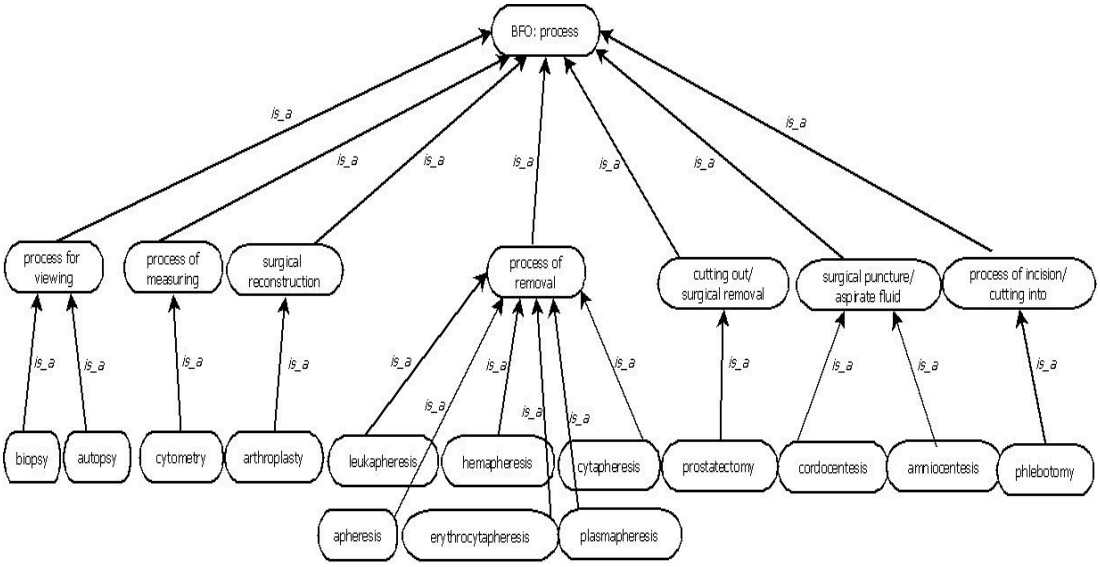
Tabela 1: Agrupamento dos termos recuperados em classes semânticas

Classe semântica	Termo recuperado (frequencia)
Process of removal	*apheresis (124), plasmapheresis (15), leukapheresis (12), hemapheresis (6), Leukapheresis (6), rythrocytapheresis (5), Plasmapheresis (3), Cytapheresis (3)*
Surgical puncture or aspirate fluid	*cordocentesis (16), amniocentesis (14), Amniocentesis (4)*
Process of incision or cutting into	*phlebotomy (32), Phlebotomy (9)*
Process of measuring	*cytometry (20)*
Process for viewing	*biopsy (7), autopsy (5)*
Surgical reconstruction	*arthroplasty (9)*
Cutting out	*prostatectomy (8)*

As classes semânticas e os termos mostrados na tabela 1 correspondem aos candidatos à ontologia sobre os processos do sangue humano. A fim de incluí-los em tal

ontologia, foi construída uma taxonomia desses elementos (vide figura 1), que representam processos hemoterapêuticos.

Figura 1: Taxonomia dos tipos de processos hemoterapêuticos



Para a construção da taxonomia partiu-se da utilização de uma ontologia de fundamentação - a *Basic Formal Ontology* (BFO) [Grenon e Smith 2004] – que representa, normalmente, uma boa prática na construção de ontologias de domínio. Assim a taxonomia inicia-se com a classe *process* da BFO e, na sequência, temos as classes correspondentes aos processos hemoterapêuticos: (i) no segundo nível, são representados os processos mais gerais, agrupados de acordo com o sufixo; e (ii) no terceiro nível, temos os processos específicos, que correspondem exatamente aos termos processuais recuperados do corpus.

5. Conclusões e trabalhos futuros

O presente artigo apresentou um estudo de caso no domínio da hemoterapia sobre a extração de termos médicos de um corpus, criado nesta área, que foram utilizados como classes de uma ontologia sobre os processos envolvidos na manipulação do sangue humano, em desenvolvimento no âmbito do *Blood Project*.

Embora se tenham atingido os propósitos aqui pretendidos, é importante ressaltar que, como pesquisa em andamento, ainda estão previstos passos como: (i) a validação das classes geradas para a ontologia, por parte de especialistas na área, com o propósito de assegurar maior representatividade dos termos; (ii) extração de termos compostos (bigramas, trigramas, etc.) para ontologia com base em técnicas estatísticas (exs: *índice de informação mútua, z-score)* e com uso de métodos de inferência; (iii) criação e processamento de um novo corpus na área de hemoterapia, que englobe as publicações científicas mais recentes na área. Tais passos são necessários para produzir resultados qualitativos mais consistentes e garantir maior qualidade à abordagem.

Como consideração final, acredita-se que as técnicas de PLN, de uma maneira geral, têm muito a contribuir com o processamento de grandes volumes de informações, tornando-o mais ágil e reduzindo drasticamente o tempo gasto por profissionais que

desempenham tarefas nesse contexto, tal como os curadores. No entanto, consideramos também que a intervenção humana é indispensável em algumas etapas da extração automática de termos para ontologias usando ferramentas de PLN, já que, atualmente, elas ainda não são capazes de tomar decisões próprias de especialistas humanos, baseando-se, exclusivamente, em informações linguísticas e estatísticas.

Referências

AABB – Advancing Transfusion and Cellular Therapies Worlwide [site] (2012). AABB ©. Disponível em: http://www.aabb.org/Pages/Homepage.aspx.

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.

Biber, D. Representativeness in Corpus Design. (1993) *Literary and Linguistic Computing*, vol. 8, n. 4.

Buitelaar, P.; Cimiano, P.; Magnini, B. (2005) "Ontology learning from text: An overview". In: Buitelaar, P.; Cimiano, P.; Magnini, B. (Ed.). *Ontology learning from text: Methods, evaluation and applications*, v.123 of Frontiers in Artificial Intelligence and Applications. IOS Press.

Ciminano, P.; Völker, J. (2005) *A framework for ontology learning and data-driven change discovery*. Institute AIFB, University of Karlsruhe, Germany.

Friedman, C.; Hripcsak G. (1999) "Natural language processing and its future in medicine". *Academic Medicine*, vol. 74, n. 8, Agosto.

Grenon, P.; Smith, B. (2004) "SNAP and SPAN: Towards Dynamic Spatial". *Spatial Cognition e Computation*, v.4, n.1, p. 69-104.

Leech, G. (1993) "Corpus annotation schemes". *Literary and Linguistic Computing* 8(4): 275-81.

McEnery, T.; Wilson, A. (2011) *Corpus Linguistics: an introduction*. Edinburgh: Edinburgh University Press, Second Edition.

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*. New York: ACM Press, p. 3–9.

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, abr.

Tognini-Bonelli, E. (2001) *Corpus Linguistics at Work*. Philadelphia: John Benjamins BV. p. 47-64.

Wächter, T.; Schroeder, M. (2010) "Semi-automated ontology generation within OBO-Edit". *BioInformatics*, vol. 26, p. 88-96.

Winnenburg, R.; Wächter, T., Plake, C.; Doms, A.; Schroeder, M. (2008) "Facts from text: can text mining help to scale-up high-quality manual curation of gene products with ontologies?" *Briefings in Bioinformatics*, vol. 8, n. 6, p. 466-478.

Aplicando *Linked Data* na publicação de dados do ENEM

Samuel Pierri Cabral[1], Nitay Batista Beduschi[1], Airton Zancanaro[2], José Leomar Todesco[12], Fernando A. O. Gauthier[12]

[1]Departamento de Informática e Estatística– Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

[2]Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

{scabral, nitay}@inf.ufsc.br, {airtonz, tite, gauthier}@egc.ufsc.br

Abstract. *The present article describes the experiment performed with the National Examination of Secondary Education (ENEM) data set, in the year of 2008, which were published in the principles of linked data. To this end, the data set were first treated in a database, performed consolidation and represented by an ontology. Then, with the use of tools, the data set were converted to a RDF format, linked to the data of DBPedia and published in a triple store. Lastly, a Web application was built to allow visualization of the data with the aid of SPARQL consultations. The experiment allowed us to establish a workflow for publication of linked geographical data, allowing of analysis and discovery of new knowledge.*

Resumo. *Este artigo descreve o experimento realizado com os dados do Exame Nacional do Ensino Médio (ENEM), do ano de 2008, que foram publicados nos princípios do Linked Data. Para tal, os dados foram primeiramente tratados em um banco de dados relacional, realizadas consolidações; e os dados, representados por uma ontologia. Em seguida, com o uso de ferramentas, foram convertidos para o formato RDF, ligados aos dados da DBPedia e publicados em um servidor de triplas. Por fim, uma aplicação Web foi construída para a visualização dos dados com auxílio de consultas SPARQL. O experimento permitiu estabelecer um fluxo para publicação de dados geográficos, possibilitando a descoberta de novos conhecimentos.*

1 Introdução

O Ministério da Educação (MEC) tem aplicado anualmente o ENEM, cujos resultados, para o governo, são tidos como um instrumento de avaliação e promoção de melhorias na educação e, para o secundarista, como a possibilidade de acesso às universidades públicas. Os dados obtidos com o exame ficam disponíveis para que a sociedade converta-os de forma tal que sejam entendidos também pelas máquinas e possam ser ligados a outros conjuntos de dados, permitindo complemento das informações.

Na tentativa de avaliar o desempenho do secundarista e a qualidade do ensino médio, o MEC criou em 1998, o ENEM (CASAGRANDE, 2009). Posteriormente, o exame passou por reestruturações e foi utilizado, também, como forma de acesso de novos estudantes a universidades públicas brasileiras através do SiSU (Sistema de

Seleção Unificada). Em 2011, segundo Carneiro (2011), foi considerado o maior exame existente na América Latina, contando com mais de 4,5 milhões de inscritos.

No ENEM, cinco competências são avaliadas, tanto nas provas objetivas como nas subjetivas (INEP, 2008). São elas: 1) dominar linguagem; 2) compreender fenômenos; 3) enfrentar situações-problema; 4) construir argumentação; e 5) elaborar propostas. Isso permite, por exemplo, que cada instituição de ensino possa criar seus próprios critérios para o acesso de novos estudantes.

Todas as informações relacionadas ao exame estão disponíveis para ser acessadas livremente através do *site* do INEP[1], que disponibiliza também, o "Manual do usuário dos microdados do ENEM". Isso permite que, através desses dados, diferentes pessoas possam acessá-los, distribuí-los, reusá-los, analisá-los ou publicá-los usando os princípios do *Linked Data*.

Este surgiu em 2006 (BERNERS-LEE, 2006) como uma alternativa para tentar resolver o problema da grande quantidade de dados disponíveis na *Web*, isto é, textos ou arquivos dos mais variados formatos, dos quais muitos só podem ser interpretados por seres humanos (BERNERS-LEE, 2010). Isso impede que aplicações (máquinas) consigam extrair informações reais contidos nesses documentos.

Assim como os *hiperlinks,* que permitem conectar documentos em um espaço único de informação global, Heath e Bizer (2011) afirmam que o *Linked Data* possibilita a ligação entre diferentes fontes de informação, formando a *Web* de dados. Isso torna possível que as aplicações genéricas operem sobre um conjunto de dados mais completo.

Pesquisas, como a de Hull (1997), discutem a integração das diferentes bases de dados, tendo o propósito de agregar mais conteúdo ao que está sendo pesquisado na *Web*. No que se refere a dados estatísticos, Zapilko e Mathiak (2011) e Kämpgen, O´Rain e Harth (2012) apresentam o formato de cubo OLAP como método para aumentar a *performance* na publicação de dados na *Web*. Já Pirrotta (2010) descreve o processo e as lições aprendidas na publicação dos dados das universidades utilizando os princípios do *Linked data*.

Para que a interligação dos dados seja possível, Bizer, Heath e Berners-Lee (2009) descrevem um conjunto de regras, conhecidas como os quatro princípios do *Linked Data*, e estabelecem um padrão para a publicação dos dados na *Web*. São elas: 1) utilizar *Uniform Resource Identifiers* (URI) para nomear as coisas; 2) usar HTTP URIs para que as pessoas possam procurar por esses nomes; 3) fornecer informações úteis utilizando os padrões *Resource Description Framework* (RDF) e *SPARQL*, quando alguém procurar por uma URI; e 4) incluir *links* para outras URIs a fim de que elas possam descobrir mais informações.

Dessa forma, as aplicações interpretam os conteúdos disponibilizados na *Web* de dados através de um modelo genérico, denominado de RDF, que se liga a outros dados no mundo na forma de triplas: sujeito, predicado e objeto (LASSILA; SWICK, 1998). A proposta do RDF é, segundo Souza e Alvarenga (2004), criar uma maneira com a qual cada página *Web*, cada recurso possam gerar sua própria metainformação, ou seja, informação sobre informação, e torná-la disponível para quem precisar.

[1] INEP: Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

Os padrões RDF, associado às ontologias e aos *namespaces* compartilhados, possibilitam combinar os dados, como no caso do ENEM, com outras fontes de dados, buscando, assim, mais informações referentes, p. ex., ao município onde foi realizada a prova: sua latitude e longitude, informações sobre seu IDH e sua população etc.

Em síntese, este trabalho tem por objetivos identificar os dados do ENEM disponíveis para serem acessados de forma aberta, tratá-los conforme a necessidade, representá-los por meio de ontologias, convertê-los no formato RDF e construir uma aplicação *Web* a fim de visualizá-los de forma amigável. Para isso, na seção 2, será apresentada a definição do escopo do trabalho; na seção 3, serão abordados o tratamento dos dados e a conversão para o formato RDF; na seção 4, são demonstrados os resultados; e, por último, na seção 5, apresentadas as considerações finais.

2 Definição do escopo do trabalho

Para a elaboração deste experimento, inicialmente procurou-se determinar quais informações relativas ao ENEM estavam disponíveis para utilização no *site* do INEP. Essa consulta foi realizada no mês de outubro de 2011, e identificou-se que, além de microdados de outros anos, o mais recente era 2008, utilizado no trabalho.

Com o auxílio do manual do usuário, percebeu-se que os dados estão estruturados em variáveis de controle do inscrito e da escola, da cidade da prova, da prova objetiva e de redação, e do questionário socioeconômico.

Dentre tantas variáveis, optou-se, para o tratamento e a publicação, agrupá-las por município em que o inscrito realizou a prova, visto que grande parte das informações nesse campo estava preenchida adequadamente. Com esse agrupamento foi possível criar indicadores das médias obtidas na prova objetiva e na redação, do sexo, da idade, da rede de ensino e da localização da escola (área urbana ou rural). Além disso, a criação de uma ontologia e o reuso de outra se fizeram necessários, permitindo que os dados do ENEM pudessem ser interligados a outros através do padrão RDF.

3 Tratamento e publicação dos dados

Os dados do ENEM de 2008 estão no formato texto e com as informações separadas por ponto e vírgula. Com o auxílio das ferramentas UltraEdit, para a conversão do arquivo no formato CSV (*Comma Separated Values*), e do MySQL Workbanch, foi possível criar as tabelas e importar os dados para o banco de dados MySQL.

De posse das informações em uma tabela no banco de dados, denominada de "CSV_ENEM", uma nova foi criada ("EN_INDICADORES_ENEM"), com os cálculos dos indicadores agrupados por município, juntamente com os outros campos descritos anteriormente, utilizados para a geração do RDF. O Quadro 1 exemplifica o tratamento realizado em um campo da tabela.

Quadro 1 – Exemplo de um campo da tabela "EN_INDICADORES_ENEM"

Campos da tabela	Descrição dos campos	Tratamento realizado
COD_MUNICIPIO	Código do IBGE do município onde foi realizada a prova	No campo ID_CIDADE_PROVA da tabela CSV_ENEM, foi utilizada parte do código para compor o código do IBGE; os municípios sem código ou com código sem equivalência aos do IBGE foram alterados para '0'.

Para atender ao escopo deste experimento, foi criada uma ontologia, denominada de "enem[2]", com o objetivo de fazer a interligação dos dados do exame a outros existentes. Essa ontologia foi criada a partir da importação da ontologia "geopoliticabr"[3], disponível no projeto Lodkem[4], que possui as propriedades referentes aos municípios brasileiros, como o código do IBGE, o nome, o *sameAs* para a DBpedia e os pontos cardeais. A ontologia "geopoliticabr" pode ser visualizada na Figura 1.

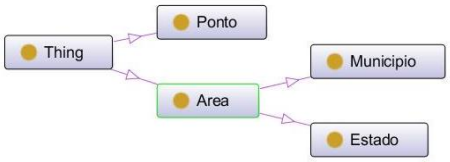


Figura 1 - Ontologia "geopoliticabr"

Com a ontologia disponível na *Web,* para ser acessada, e os dados tratados em uma tabela de banco de dados, foi possível construir um mapeamento entre eles, de forma tal, que o *software* D2RQ tivesse condições de gerar o RDF. Isso permitiu que, importando o RDF para um servidor de triplas (OpenLink Virtuoso), os dados se tornassem disponíveis, ligados a outros e a DBPedia, possibilitando a consulta através da linguagem SPARQL e a construção de uma aplicação.

4 Demonstração dos resultados

Para facilitar a visualização dos resultados, uma aplicação *Web* foi criada e disponibilizada no *site* http://www.lodkem.ufsc.br:8080/enem, com a finalidade de construir gráficos dinâmicos, utilizando as consultas SPARQL. A consulta de um município é exemplificada no Quadro 2.

Quadro 2 - Consulta SPARQL do município de Florianópolis

```
PREFIX geo: <http://lodkem.ufsc.br/onto/geopoliticabr#>
PREFIX enem: <http://lodkem.ufsc.br/onto/enem#>
SELECT DISTINCT ?nome ?codibge ?lat ?lon ?area ?totalaluno
WHERE {
  ?x geo:temNomeMun ?nome .
  ?x geo:temCodIbgeMun ?codibge .
  ?x geo:temPontoCentralMun ?p .
  ?x geo:temAreaMun ?area .
  ?p geo:lat ?lat .
  ?p geo:lon ?lon .
  ?x enem:temTotalAluno
  ?totalaluno .
  filter (?codibge != 0 && regex(?nome,'florianopolis', 'i'))
} ORDER BY (?codibge)
```

Através dessa consulta, os dados do município, como o nome, o código do IBGE e a latitude e longitude, são buscados através da ontologia "geopoliticabr". Já o total de alunos são oriundos da ontologia "enem". Além disso, o filtro "regex" permite trazer todos os municípios com o nome informado, e a *string* 'i' significa *case insensitive*. O campo e os dados geográficos da cidade pesquisada podem ser observados na Figura 2.

[2] Disponível em: http://lodkem.ufsc.br/onto/enem.owl
[3] Disponível em http://lodkem.ufsc.br/onto/geopoliticabr.owl
[4] http://lodkem.egc.ufsc.br/

Para gerar o gráfico com as médias das cinco competências avaliadas na prova objetiva e a média geral, é executada a seguinte consulta (Quadro 3):

Quadro 3 - Consulta SPARQL com a média das competências

```
PREFIX enem: <http://lodkem.ufsc.br/onto/enem#>
PREFIX geo: <http://lodkem.ufsc.br/onto/geopoliticabr#>
SELECT DISTINCT ?nome ?codibge ?compobj1 ?compobj2 ?compobj3 ?compobj4
?compobj5 ?mediageralobj
WHERE {
   ?x geo:temNomeMun ?nome .
   ?x geo:temCodIbgeMun ?codibge .
   ?x enem:temMediaCompetenciaObj1 ?compobj1 .
   ?x enem:temMediaCompetenciaObj2 ?compobj2 .
   ?x enem:temMediaCompetenciaObj3 ?compobj3 .
   ?x enem:temMediaCompetenciaObj4 ?compobj4 .
   ?x enem:temMediaCompetenciaObj5 ?compobj5 .
   ?x enem:temMediaGeralObj ?mediageralobj .
   filter (?codibge = 4205407 ) }
```

Essa consulta tem como resultado o gráfico apresentado na Figura 3. Vale ressaltar que o foco deste trabalho não é analisar os resultados cognitivos dos estudantes, e sim, a disponibilização e visualização das informações.

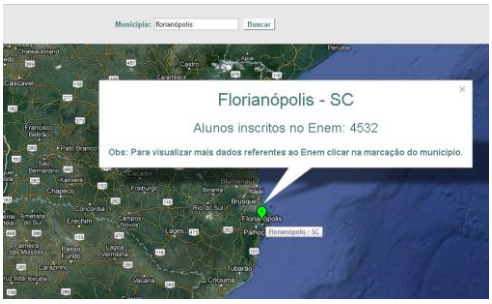


Figura 2 - Mapa com o resultado da consulta por município

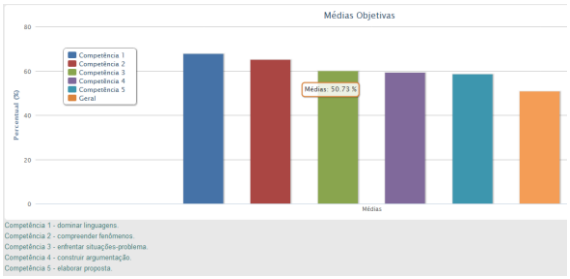


Figura 3 - Média das cinco competências da prova objetiva e média geral

Desta forma, percebe-se que, com a ligação dos dados a outras fontes, é possível descobrir facilmente mais informações, ampliando o conhecimento sobre determinado assunto.

5 Considerações finais

Este trabalho buscou identificar os dados do ENEM que estão disponíveis de forma aberta, fazer o seu tratamento, construir e reusar ontologias, convertê-los no formato RDF e desenvolver uma aplicação *Web* que lhes permitisse a visualização de uma forma amigável.

Quanto às lições aprendidas, concernentes à publicação de dados abertos, viu-se que, de um modo geral, o governo vem disponibilizando uma quantidade significativa de dados. Entretanto, a falta de estruturação e dados incompletos dificultaram a sua manipulação de forma adequada.

Para trabalhos futuros, será realizada a extensão da ontologia "geopoliticabr", incorporando novos conceitos e a inclusão de dados do ENEM dos outros anos. Além disso, para aumentar a escala de publicação e análise através de séries históricas dos dados, o vocabulário *Data Cube* RDF está sendo automatizado.

Dessa forma, a contribuição deste experimento está na utilização de métodos e ferramentas para a publicação de dados utilizando os princípios do *linked data.* Acredita-se que, com infraestrutura de acesso aos dados padronizados e em escala global, será possível torná-los disponíveis tanto para o uso humano quanto para as máquinas. Isso facilitará para que as aplicações construídas tenham condições de reutilizar os dados facilmente, formando uma base fundamentada para novos conhecimentos.

Referências

BERNERS-LEE, Tim. **Linked Data: Design Issues.** 2006. Disponível em: <http://www.w3.org/DesignIssues/LinkedData.html>. Acesso em: 05 jul. 2012.

_____. Long Live the Web: A Call for Continued Open Standards and Neutrality. **Scientific American**, Dezembro 2010. Disponível em: <http://www.scientificamerican.com/article.cfm?id=long-live-the-web&page=6>. Acesso em: 29 Abr. 2012.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked Data - The Story So Far**. Int. Journal On Semantic Web And Inf. Systems (ijswis),** p. 1-22. mar. 2009.

CARNEIRO, V. L. Políticas Públicas Educacionais e Gestão do Ensino Médio no Brasil: o Exame Nacional de Ensino Médio - ENEM e suas implicações para o trabalho docente. In: **XXV Simpósio Brasileiro e II Congresso Ibero-Americano de Política e Administração da Educação**, 2011, São Paulo.

CASAGRANDE, A. L. Avaliação: a redefinição do papel do ENEM. In **XI Seminário Estadual da ANPAE-SP**, 2009.

INEP. **Matriz De Referência Para O Enem**. Instituto Nacional De Estudos E Pesquisas Educacionais Anísio Teixeira. Brasília, p. 26. 2008.

HEATH, T.; BIZER, C. . **Linked Data: Evolving the Web into a Global Data Space**: Morgan & Claypool, 2011.

HULL, R. "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective," in **ACMSymposium on Principles of Databases**, 1997, pp. 51–61.

KÄMPGEN, B.; O'RAIN, S.; HARTH, A. **Interacting with Statistical Linked Data via OLAP Operations. in** Inter. Workshop on Interacting with Linked Data. 2012.

LASSILA, O.; SWICK, R. R. **Resource Description Framework (RDF) Model and Syntax Specification**. Disponível em: <http://www.w3.org/1998/10/WD-rdf-syntax-19981008>. Acesso em: 28 ago. 2012.

PIRROTTA, G. Linking Italian University statistics. **ACM International Conference Proceeding Series**, 2010. Graz.

SOUZA, R. R. ; ALVARENGA, L. A Web Semântica e suas contribuições para a ciência da informação. **Ciência da Informação**, Brasília, v. 33, n. 1, p. 132-141, 2004.

ZAPILKO, B.; MATHIAK, B. Performing Statistical Methods on Linked DataProc. Int'l Conf. on Dublin Core and Metadata Applications 2011. Anais...2011.

Modelagem de relações conceituais para a área nuclear

Luana Farias Sales[1,2] , Luís Fernando Sayão[2], Dilza Fonseca da Motta[3]

[1]Programa de Pós-Graduação em Ciência da Informação – Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT) e Universidade Federal do Rio de Janeiro (UFRJ)

[2]Comissão Nacional de Energia Nuclear (CNEN)

[3]Financiadora de Estudos e Projetos (FINEP)

lsales@ien.gov.br; lsayao@cnen.gov.br;
dilzafmotta@yahoo.com.br

Abstract. *The nuclear energy area is a complex domain involving a large number of disciplines, concepts and relations. Despite its long tradition in organizing, processing and dissemination of information, reflected in important databases and international information systems, in recent decades this field has not evolved satisfactorily regarding the development of tools to standardize terminology and, consequently, concerning the extension of the theoretical and methodological framework for conceptual modeling. This fact creates an obstacle in the development of more sophisticated information systems. Starting from the systematization of ontic conceptual relations in the domain of the nuclear area, this paper presents a new way to conceptual modeling, anchored in the theoretical basis of information science. The proposed model, based on a classification principles, seeks to combine categorical and formal relations, to reaching the triadic model of relations for the nuclear area.*

Resumo. *A área de Energia Nuclear é um domínio complexo que envolve um grande número de disciplinas, conceitos e relações. Apesar da sua longa tradição na organização, no tratamento e na disseminação de informação, refletido em importantes bases de dados e sistemas internacionais de informação, nas últimas décadas não evoluiu a contento no que diz respeito à elaboração de instrumentos de padronização terminológica e, consequentemente, no que diz respeito à ampliação de arcabouço teórico-metodológico para modelagem conceitual. Esse fato cria um obstáculo no desenvolvimento de sistemas de informação mais sofisticados. Partindo da sistematização das relações conceituais ônticas existentes no domínio da área nuclear, o presente trabalho visa apresentar um novo caminho para a modelagem conceitual, ancorado na base teórica da Ciência da Informação. A modelagem proposta, baseada em princípios classificatórios, procura combinar relações categoriais e formais, chegando ao modelo triádico de relações para a área de ciências nucleares.*

1. Considerações iniciais

A necessidade de elaborar modelos deriva inicialmente da dificuldade de o homem entender a complexidade da realidade do universo que o envolve. Assim, em uma primeira instância, o ser humano elabora modelos para compreender o mundo ou simplesmente uma questão no mundo; estabelecer padrões de comunicação entre ele e outros seres e representar de forma simplificada um objeto ou uma situação no mundo.

Os modelos podem ser construídos "por meio de formalismos matemáticos, fenomenológicos ou conceituais" e permitem "testar hipóteses, tirar conclusões, caminhar no sentido da generalização e da particularização, através de processos de indução e têm sempre vida provisória". (SAYÃO, 2001, p.83).

Os modelos conceituais são construídos a partir de abstrações que especificam relacionamentos entre conceitos, trabalhando semelhanças, diferenças e outras associações de significado.

Na esfera da modelagem conceitual, identificar os tipos de relações presentes em um domínio pode ser considerada uma etapa da modelagem conceitual deste domínio, pois são as relações que ligam os conceitos a outros conceitos, permitindo evidenciar a abstração de uma dada realidade.

O problema que se coloca, no entanto, é que as relações não-hierárquicas, também chamadas de associativas ou ônticas se manifestam diferentemente de acordo com a área, variando de acordo com o objetivo e com a questão que se visa modelar. Acredita-se que uma possível solução para este problema esteja na proposta de um método para modelagem de relações conceituais que possa ser generalizável a qualquer domínio. O presente trabalho tem por objetivo apresentar um método para modelagem de relações não-hierárquicas que foi aplicado, neste momento, na área de ciências nucleares.

2. A área de Energia Nuclear e a necessidade de modelos conceituais

No domínio da Ciência da Computação, a modelagem conceitual é um estágio anterior ao desenvolvimento do sistema. Nesta área, a elaboração de modelos conceituais fornece subsídios para construção de sistemas eficazes aos seus propósitos. Já na Ciência da Informação, os modelos são construídos para servirem de instrumentos padronizadores de informações, tornando a recuperação e a comunicação mais precisas.

A área de Energia Nuclear tem uma longa tradição na organização, tratamento e na disseminação da informação. O International Nuclear System (INIS), órgão subordinado à Agência Internacional de Energia Atômica, da ONU, deu prosseguimento à política de valorização da informação nuclear, como insumo estratégico para o desenvolvimento das aplicações pacíficas da energia nuclear.

No que tange a elaboração de regras e padrões, o INIS criou um conjunto de ferramentas terminológicas, incluindo classificação e tesauro. O desenvolvimento do tesauro foi derivado de um modelo gráfico chamado INIS Terminology Charts, que exibia representações gráficas dos conceitos envolvidos e as suas relações, incluindo a intensidade de cada relação e o seu tipo (INIS, 1970).

As iniciativas de construtos teminológicos do INIS foram muito interessantes e consideradas avançadas para os padrões da época. No entanto, apesar dos indícios da

necessidade desses instrumentos acompanharem evolutivamente a produção do conhecimento e as mudanças temporais, a área de Energia Nuclear, nos últimos anos, se manteve indiferente em relação à atualização metodológica de seus instrumentos. Enquanto na tabela de classificação houve uma diminuição considerável do número de áreas abrangidas, o tesauro vem se mantendo estático no que tange a inserção de novos termos e novas relações.

Mesmo com a descontinuidade dessas ferramentas conceituais, elas continuam sendo de fundamental importância para essa área, principalmente devido ao fato dela sempre ter utilizado, para compreensão de seus fenômenos, modelos e generalizações como uma ponte entre os níveis de observação e o teórico. Nas últimas décadas, o uso massivo de técnicas de simulação em computadores para realização de experimentos na área, mais uma vez, coloca em voga a necessidade dos modelos. Por fim, a intensa geração de dados, fruto do uso de *software*, leva à necessidade de curadoria desses dados para fins de recuperação e reuso e, neste caso, são os modelos que dão estrutura e significado aos dados. Esses fatores evidenciam a necessidade de modelos conceituais e consequentemente da modelagem das relações conceituais.

3. A modelagem das relações da área de Energia Nuclear

A modelagem das relações da área de Energia Nuclear foi realizada a partir da base teórica da Ciência da Informação para construção de linguagens ou modelos (CAMPOS, 2001b). Essa escolha se justifica por ter esta área abordagens teórico-metodológicas consolidadas para construção de instrumentos de representação da informação e do conhecimento. Conforme abordagem de Campos (2001a), a união da tríade teórica formada pela Teoria da Classificação Facetada (RANGANATHAN, 1967), Teoria Geral da Terminologia (WÜSTER, 1981) e Teoria do Conceito (DAHLBERG, 1978) vem atendendo não apenas a elaboração de linguagens, mas também toda e qualquer necessidade de classificação, sistematização, mapeamento de domínio e modelagem conceitual.

Em trabalho apresentado anteriormente (SALES, 2008), um modelo triádico de relações foi proposto para reunir as relações da Ciência da Informação (duplas de categorias, ex: coisa-material, as quais são aqui chamadas de **Relações Categoriais**) e também as da Ciência da Computação (**Relações formais**, ex: has_material, que revelam a forma como um conceito se relaciona com outro). Este modelo é composto da seguinte forma: <Relação categorial1 – relação formal – Relação categorial2> = (<rc1- rf- rc2>). Ex: <Processo-results-Entidade>

Para se chegar ao modelo triádico de relações foi utilizado no processo de modelagem um método que reuniu aspectos conceituais, lógico-classificatórios e matemáticos. Este método está sendo chamado *a priori* de "Método relacional-categorial", pois visa estabelecer relações a partir da combinação das categorias existentes no domínio mapeado. No que tange aos aspectos conceituais, o método se valeu de abordagens advindas da Teoria do Conceito e da Terminologia para a identificação de definições terminológicas consistentes para os termos que fizeram parte do corpus selecionado, bem como para a intervenção nessas definições, quando necessário. Quanto aos aspectos lógico -classificatórios, o método se valeu da Teoria da Classificação Facetada para identificação das categorias e classificação dos termos identificados para compor o corpus. Com relação aos aspectos matemáticos, o método

utilizou-se de arranjo em uma análise combinatória para compor os pares de relações categoriais.

O método abrangeu as seguintes etapas: 1)Análise das definições. 2) Categorização dos termos; 3)Análise combinatória das categorias; 4)Retirada de assertivas [1]das definições no modelo <rc1-rf-rc2>; 5)Categorização das assertivas e identificação das relações formais; 6)Sistematização das relações formais possíveis para cada par de relações categoriais.

A primeira etapa do método foi a análise do Glossário Nuclear da CNEN (2011), composto por 107 termos, que se configuraram como corpus. Suas definições bem construídas auxiliaram na categorização dos termos e, consequentemente, no estabelecimento das relações entre os termos.

A segunda etapa do método foi baseada na Teoria da Classificação Facetada de Ranganathan que sugere cinco categorias fundamentais representadas pela sigla PMEST, cujas iniciais significam: Personalidade, Matéria, Energia, Espaço e Tempo[2]. Os termos foram separados em categorias baseadas nestas últimas, adaptadas para a área em análise.

Chegou-se então a sete categorias: Entidade, Equipamento, Propriedade, Matéria, Processo, Espaço, Tempo. Na categoria Entidade foram agrupados todos os objetos, indivíduos, agentes e produtos. Na categoria Equipamento foram considerados os instrumentos, as ferramentas, as máquinas ou outros artefatos, aparelhos, utensílios, dispositivos e aparatos experimentais que sirvam para executar um processo e/ou gerar um produto. Os conceitos referentes às características, atributos, medidas, dimensões e outras qualidades dos objetos, das entidades, dos processos, do tempo, do espaço etc. foram incluídos na categoria Propriedade. Os conceitos referentes aos materiais constituintes e substâncias foram categorizados em Matéria. Ações, operações e fenômenos em geral foram agregados na categoria Processo. Os conceitos referentes às áreas, locais ou regiões foram agrupados na categoria Espaço. Por fim, na categoria Tempo foram incluídos os conceitos cuja noção de tempo é determinante. Veja o quadro 1.

Quadro 1 - Exemplo de categorização dos conceitos constantes nos glossários

ENTIDADE	EQUIPAMENTO	PROPRIEDADE	MATÉRIA	PROCESSO	ESPAÇO	TEMPO
Átomo	Acelerador	Radioatividade	Elemento combustível	Radioterapia	Área controlada	Meia vida

[1] Assertivas são afirmações ou proposições que podem ser atestadas como verdadeiras. Neste caso, as assertivas são modeladas por triplas que vem a ser predicados binários.

[2]Para Ranganathan, o PMEST pode ser explicado da seguinte forma: Na categoria **Tempo** estão as **i**déias isoladas de tempo, na categoria **Espaço** estão aquelas referentes ao local de pertencimento de um determinado objeto, seja ele indivíduo, coisa, fenômeno, entre outras entidades. Na categoria **Energia** estão as idéias de processo, ação ou fenômeno. Na categoria **Matéria** - suas manifestações são de duas espécies - Material e Propriedade, que são partes intrínsecas de um objeto ou processo. Na categoria **Personalidade**, Ranganathan considera através de um método residual, tudo aquilo que não cabe nas outras categorias.

A terceira etapa constou da análise combinatória das categorias identificadas anteriormente. Como na Ciência da Informação as relações se manifestam entre duplas de categorias, combinar as categorias que mapeiam uma área é uma forma de identificar, no nível mais genérico possível, as possibilidades de relações a serem encontradas em um domínio. A partir da análise combinatória chegou-se a 49 pares de relações categoriais. Para citar algumas: entidade-entidade, entidade–propriedade, entidade–matéria, entidade–equipamento, entidade–processo, entidade–tempo, entidade–espaço, entre outras.

Para encontrar as relações formais, partiu-se então para a quarta etapa do método que constou da retirada de assertivas no modelo relação categorial1-relação formal-relação categorial2 <rc1-rf-rc2>, a partir da análise das definições. Conforme primeira coluna do quadro 2:

Quadro 2 - Identificação e categorização das assertivas

ASSERTIVA IDENTIFICADA	CATEGORIZAÇÃO DAS ASSERTIVAS
Área controlada- controla -exposições à radiação	Espaço - controla - processo
Área controlada –previne - disseminação e contaminação radioativa	Espaço – previne- processo

A quinta etapa constou da categorização das assertivas, reescrevendo-as de acordo com as categorias identificadas na etapa 2. O objetivo foi generalizar, no grau máximo, as relações categoriais, deixando em um nível mais específico apenas as relações formais, já que a ideia era identificar todas as possibilidades de relações formais do domínio escolhido. Esta etapa pode ser observada na segunda coluna do quadro 3.

Na sexta etapa, foi realizada a sistematização das relações formais possíveis para cada par de relações categoriais. Assim, chegou-se a 19 pares de categorias e 39 relações formais. No quadro 3 pode ser visualizada uma pequena amostra das relações identificadas.

Quadro 3 - Sistematização das relações

RELAÇÃO CATEGORIAL	RELAÇÃO FORMAL	MODELO TRIÁDICO
ENTIDADE – PROCESSO	emite	entidade - emite- processo
ENTIDADE - ENTIDADE	parte_de contém transforma	entidade - parte_de - entidade entidade – contém - entidade entidade - transforma - entidade
ENTIDADE - PROCESSO	recebe sofre	entidade - recebe - processo entidade – sofre - processo
ENTIDADE - PROPRIEDADE	representa	entidade – representa - propriedade
EQUIPAMENTO - ENTIDADE	acelera produz aumenta	equipamento – acelera - entidade equipamento - produz - entidade equipamento – aumenta - entidade
EQUIPAMENTO -MATÉRIA	contém reutiliza utiliza	equipamento – contém - matéria equipamento – reutiliza - matéria equipamento – utiliza - matéria

5. Considerações finais

O presente trabalho é fruto de estudos que caminham em direção à busca dos fundamentos de modelagem. O assunto em questão é abordado aqui como um ponto de

interseção entre a Ciência da Informação e a Ciência da Computação, pois reúne relações interessantes para as duas áreas. A modelagem conceitual requer cuidado especial com conceitos e também com as relações entre esses conceitos. Desta forma, este trabalho contemplou apenas um desses aspectos: Relações conceituais ônticas para modelagem.

Este trabalho é parte de um projeto maior que está em andamento, que envolve curadoria digital de dados de pesquisas nucleares, o que pressupõe a necessidade de modelo conceitual e consequentemente uma modelagem das relações. Entende-se que a modelagem das relações conceituais será de fundamental importância para ligar os dados de pesquisa aos documentos que os geraram. Sendo assim, pretende-se também, em momento oportuno, realizar um experimento utilizando o modelo de relações identificadas para, em uma publicação ampliada, ter os dados científicos ligados às publicações de forma consistente.

6. Referências

CAMPOS, Maria Luiza de Almeida. **A organização de unidades do conhecimento em hiperdocumentos: um modelo conceitual como um espaço comunicacional para a realização de autoria**. Rio de Janeiro: IBICT/UFRJ, 2001b. Tese de Doutorado.

CAMPOS, Maria Luiza de Almeida. **Linguagem documentária**: teorias que fundamentam sua elaboração. Niterói: EdUFF, 2001a.

CNEN. **Glossário de termos usados em Energia Nuclear**. 2011. Disponível em: <www.cnen.gov.br/noticias/documentos/**glossario**_tecnico.pdf>. Acesso em 26 jun. 2012.

DAHLBERG, I. **Ontical structures and universal classification**. Bangalore: Sarada Ranganathan Endowment, 1978.

GUEDES, V; BORSCHIVER, S. **Bibliometria**: uma ferramenta estatística para a gestão da informação e do conhecimento, em sistemas de informação, de comunicação e de avaliação científica e tecnológica, 2005. Disponível em: <http://dici.ibict.br/archive/00000508/01/VaniaLSGuedes.pdf>. Acesso em: 12 jul 2012.

INIS: **Terminology charts**. Viena: IAEA, 1970.

RANGANATHAN, S.R. **Prolegomena to library classification**. Bombay: Asia Publishing House, 1967.

SALES, Luana Farias. Modelo triádico de relações para aplicação em ontologias. Seminário de pesquisa em ontologia no Brasil. In: SEMINÁRIO BRASILEIRO DE ONTOLOGIAS, 1., 11 e 12 de Julho 2008, Niterói, Rio de Janeiro. **Anais...** Niterói: UFF, 2008. Disponível em: <http://www.uff.br/ontologia/artigos/13.pdf> Acesso em: 26 jun 2012.

SAYÃO, Luis Fernando. Modelos teóricos em Ciência da Informação: abstração e método científico. **Ciência da Informação**, Brasília: IBICT, v. 30, n. 1, p. 82-91, jan./abr. 2001.

WÜSTER, E. L'étude scientifique qénérale de laterminologie, zone frontalière entre la linguistique, la logique, l'ontologie, L'informatique et les sciences des chose. In: RONDEAU, G. ; FELBER, E. (Org.). **Textes choisis de terminologie**. Québec: GIRSERM, 1981. V.I : fondéments théoriques de la terminologie, p. 57-114.

Ontologia Probabilística para Auxiliar na Recuperação de Modelos Biológicos[1]

Wladimir Pereira, Kate Revoredo

Programa de Pós-Graduação em Informática

Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

Av. Pasteur, 296 – Urca – Cep 22290-240 – Rio de Janeiro – RJ – Brazil

{wladimir.pereira, katerevoredo}@uniriotec.br

Abstract. *The Cell Component Ontology (CelO), an ontology expressed in OWL-DL that describes semantically biological models associated with the context of electrophysiology, has no support for dealing with uncertainty. It is demonstrated in this paper that a computational environment based on ontologies (CelO) and Bayesian Networks can help researchers in the modeling phase of the cycle of experimental knowledge of Biology, retrieving accurately biological models.*

Resumo. *A Cell Component Ontology (CelO), uma ontologia expressa em OWL-DL que possibilita expressar a semântica de modelos biológicos associados ao contexto da eletrofisiologia, não possui suporte para lidar com a incerteza. É demonstrado neste trabalho que um ambiente computacional baseado em ontologias (CelO) e Redes Bayesianas é capaz de auxiliar o pesquisador na fase de modelagem do ciclo experimental de conhecimento da Biologia, recuperando modelos biológicos de uma maneira mais precisa.*

1. Introdução

Em [Matos et al. 2010] foi apresentada a Cell Component Ontology (CelO), uma ontologia expressa em OWL-DL que é derivada da CellML [Cuellar et al. 2003], uma linguagem de marcação baseada em XML (eXtensible Markup Language) [Bray et al. 2000] criada especificamente com o propósito de descrever variáveis, equações e componentes de modelos biológicos de maneira formal, sem ambiguidades, legível por humanos e processável por máquinas.

Cada modelo CellML é composto por uma rede de componentes interconectados, que é a menor unidade funcional do modelo, e por variáveis, que são entidades que têm como propósito representar quantidades usadas nas equações. Além disso, há as conexões, que mapeiam variáveis entre componentes, permitindo a troca de informações entre eles. A Figura 1 mostra um esquema dos elementos que compõem um modelo CellML.

[1] Esse trabalho faz parte do escopo do projeto "Infraestrutura de apoio a Gerência de experimentos científicos em Modelagem Computacional" com apoio do CNPQ (número 559998/2010-4)

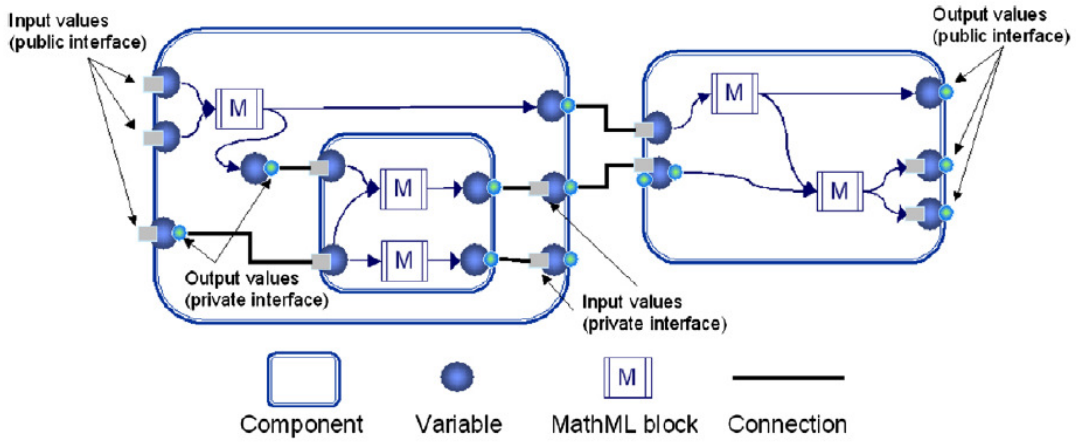


Figura 1. Representação em esquema dos elementos de um modelo CellML [Matos et al. 2010]

O objetivo da ontologia CelO é acrescentar semântica a modelos biológicos descritos em CellML, associados ao contexto da eletrofisiologia, possibilitando expressar o conhecimento intrínseco do modelo, possibilitar a validação semântica de novos modelos, reusar componentes de outros modelos, automatizar processos de composição de modelos e possibilitar que a procura de modelos seja realizada de forma semântica.

A integração da ontologia CelO com a CellML possibilita que o pesquisador modele em um nível alto de abstração e execute computacionalmente o modelo sem necessidade de conhecimento da linguagem em XML.

De acordo com [Macedo 2005], o ciclo experimental do conhecimento da Biologia passa por três fases, que podem ser vistas na Figura 2: na primeira, modelos biológicos são propostos e hipóteses são apresentadas; na segunda, simulações computacionais são executadas com os modelos biológicos propostos, combinando dados de diferentes experimentos físicos, gerando predições sobre o comportamento do sistema, provendo uma visão mais acurada dos fenômenos estudados; na terceira, o resultado de cada simulação é analisado, podendo surgir novas hipóteses desta análise, o que reiniciaria o ciclo.

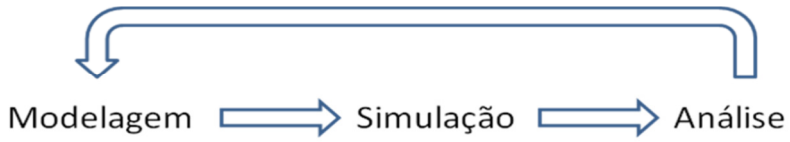


Figura 2. Ciclo Experimental do Conhecimento da Biologia [Macedo 2005]

Na fase de modelagem, que é o foco deste trabalho, o pesquisador pode obter na ontologia CelO a representação semântica do conceito ou fenômeno de interesse (por exemplo, o "potencial da membrana" e "canal iônico de sódio") e pesquisar quais modelos biológicos estão de alguma forma associados ao conceito ou ao fenômeno pesquisado. Em seguida, o pesquisador pode escolher um dos modelos biológicos listados para executar as simulações.

Dentro deste ciclo, a etapa de recuperação de um modelo biológico a ser tomado como ponto de partida deve ser precisa e retornar o modelo biológico mais adequado à necessidade do pesquisador, já que novos modelos biológicos são desenvolvidos a partir

de componentes de um modelo biológico existente. Um novo componente pode ser inserido e o modelo biológico ajustado, estabelecendo a conexão deste com os demais componentes. Após a simulação, dependendo dos resultados obtidos, a inclusão deste novo componente é confirmada ou o mesmo é substituído. Este processo pode se repetir por diversas vezes, o que torna o processo trabalhoso e sujeito a erros.

A CelO não possui suporte para lidar com a incerteza, ou seja, não é possível definir um grau intermediário de pertinência dos modelos biológicos existentes no repositório à consulta realizada. Como exemplo, ao pesquisar por "potencial da membrana" e "canal iônico de sódio", o agente responsável pela pesquisa, caso não consiga encontrar uma resposta categórica, deveria agir com um grau de incerteza, informando os modelos biológicos com maior probabilidade de atender às necessidades do pesquisador.

Por outro lado, a pesquisa feita por Ding e Peng [2004] e o trabalho de Ding et al. [2006], que gerou a linguagem BayesOWL, tiveram o objetivo de estender a OWL para representar a incerteza por meio do uso de redes bayesianas [Charniak 1991]. Os autores apresentam o conceito de probabilidade dentro da OWL, isto é, a semântica da OWL é ampliada através de marcações adicionais visando representar a incerteza. O resultado é uma ontologia que pode ser traduzida em uma rede Bayesiana, porém, em ambos os casos, o uso de anotações particulares do domínio limitam a capacidade de expressar modelos probabilísticos mais complexos ou genéricos, restringindo as soluções para classes de problemas muito específicos. No caso da BayesOWL, o foco é o mapeamento de ontologias, desta forma, a estrutura da linguagem é adequada para que este objetivo seja alcançado.

Visando a interoperabilidade com ontologias não probabilísticas, a linguagem PR-OWL foi proposta por [Costa e Laskey 2006]. A linguagem também é uma extensão para a linguagem OWL e o modelador pode obter uma ontologia em OWL padrão e utilizar os recursos da PR-OWL apenas para as partes da ontologia que necessitarem de suporte probabilístico. Em sua abordagem, ontologias OWL podem ser usadas para representar modelos probabilísticos complexos, de uma forma que é suficientemente flexível para ser usado por diversas ferramentas probabilísticas baseadas em redes Bayesianas. O problema desta abordagem é que, para lidar com a incerteza, é necessário modificar e reorganizar a base de conhecimento original, através da introdução de novas relações. Tarefa esta que pode ser trabalhosa e normalmente requer um bom conhecimento em redes Bayesianas. Além disso, requer a participação de um especialista para criar as tabelas de probabilidades condicionais.

Em [Devitt et al. 2006], os autores apresentam um algoritmo para automatizar a construção de Redes Bayesianas e representar com precisão um domínio de interesse. As tarefas envolvidas neste processo exigem a introdução de um especialista na definição de quais propriedades da ontologia ou quais relações entre os conceitos correspondem aos relacionamentos da rede bayesiana. É uma abordagem muito interessante, porque as dependências entre os nós que correspondem as classes da ontologia que não estão explicitadas na ontologia podem ser identificadas por este especialista. A tarefa de estimar as probabilidades condicionais não foi tratada nesse trabalho.

O objetivo deste trabalho é demonstrar que um ambiente computacional baseado em ontologias (CelO) e Redes Bayesianas é capaz de auxiliar o pesquisador na fase de

modelagem do ciclo experimental de conhecimento da Biologia, recuperando modelos biológicos de uma maneira mais precisa.

2. Proposta

Conforme pôde ser visto nos trabalhos citados na seção anterior, uma abordagem frequentemente utilizada para a gestão do conhecimento e da incerteza é a combinação de Ontologias e Redes Bayesianas.

Rede Bayesiana (RB) é um grafo direcionado acíclico, onde cada nó é uma variável identificada a partir do domínio de aplicação e cada arco representa a dependência direta entre as variáveis. Cada variável tem um domínio de valores possíveis que ela pode assumir e associada a ela há uma tabela de probabilidades condicionais (CPT) que fornece a probabilidade para cada valor possível desta variável [Charniak 1991].

A Figura 3 mostra um exemplo de RB onde é possível perceber que a variável *dog-out* é influenciada diretamente tanto pela variável *family-out* como pela variável *bowel-problem* e que a mesma possui uma CPT associada a ela que pode ser definida como P(*dog-out*) = <0.99, 0.90, 0.97, 0.30>.

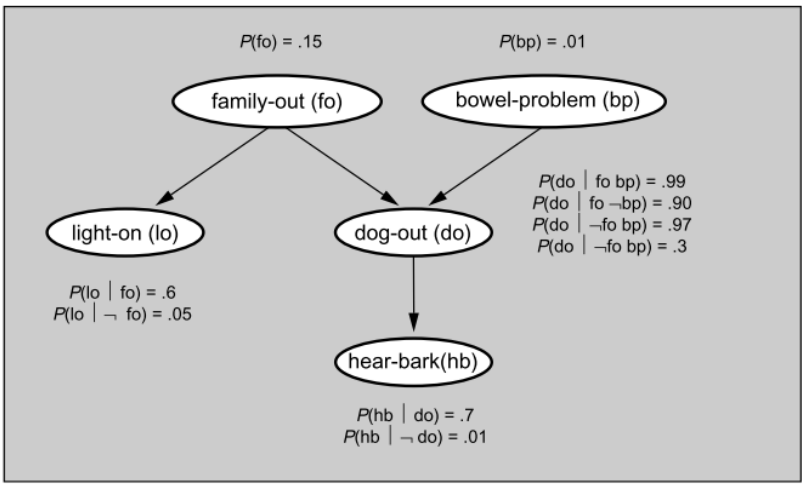


Figura 3. Exemplo de Rede Bayesiana [Charniak 1991]

O uso de ontologias foi descrito em [Guarino 1995] como um meio para adicionar semântica à web. Ele define ontologias como uma representação formal de um conhecimento compartilhado, processável por máquinas. Uma ontologia representa as classes de entidades de um domínio de aplicação, as propriedades das classes, as relações entre as classes e os papéis que as classes podem desempenhar.

O conhecimento pode ser extraído de uma ontologia usando o raciocínio lógico, explorando as relações entre as classes (conceitos) e os fatos armazenados nele (as instâncias das classes). Isto é, ontologias consistem em duas partes: uma parte referida como TBox, que contém o conhecimento sobre os conceitos (classes, por exemplo) e as relações entre eles (ou seja, papéis); e uma outra parte referida como ABox, que contém conhecimento sobre as entidades (ou seja, indivíduos) e como eles se relacionam com as classes [Andrea e Franco 2011].

Segundo [Devitt et al. 2006], a tarefa de construção da estrutura da RB é dependente do conhecimento de um especialista e possui as seguintes etapas:

1. Identificar os conceitos relevantes definidos no TBox da ontologia e mapear cada um deles como uma variável da RB.
2. Especificar os valores possíveis para cada uma destas variáveis.
3. Identificar as relações de influência entre as variáveis.

A etapa de obtenção dos parâmetros das distribuições de probabilidade para cada variável (as CPTs) consiste na aprendizagem das distribuições de probabilidade inicial, que são calculadas diretamente das instâncias de ontologia (ABox).

A ideia é que a RB gerada após estas etapas represente o conhecimento probabilístico codificado por uma ontologia tanto em nível de conceito como em nível de instância e, quando associado à ontologia CelO, torne a recuperação de modelos biológicos mais precisa, o que auxiliará o pesquisador na fase de modelagem.

3. Considerações Finais

Neste trabalho é proposta uma abordagem que visa auxiliar o pesquisador na fase de modelagem do ciclo experimental de conhecimento da Biologia, recuperando modelos biológicos de uma maneira mais precisa. Além de detalhar a proposta, foram apresentados os conceitos de RB e de Ontologias, além de trabalhos relacionados ao tema.

Ao contrário de algumas das pesquisas citadas, esta abordagem tem como grande vantagem o fato de existir uma separação entre o conhecimento do domínio e o conhecimento probabilístico, isto é, os conceitos de probabilidade não são representados dentro da ontologia e a base de conhecimentos não é alterada. Desta forma, a proposta não exige que a OWL seja estendida.

Além disso, consideramos a abordagem proposta neste artigo mais vantajosa em um contexto geral já que propõe aprender uma RB a partir das instâncias da ontologia, diminuindo a necessidade de um especialista na definição das distribuições de probabilidade condicional.

Para a avaliação da proposta, será realizado um experimento, utilizando um repositório de modelos biológicos representados através da CelO, com foco no processo de recuperação de modelos. Visando confirmar o ganho da proposta, serão comparados os resultados obtidos com os apresentados em [Matos et al. 2010].

Referências

Andrea, B., e Franco, T. (2011). Mining Bayesian networks out of ontologies. Journal of Intelligent Information Systems. Published online first, 13 June 2011. doi:10.1007/s10844-011-0165-4.

Bray, T., Paoli, J. e Sperberg-McQueen, C. M. (2000). Extensible Markup Language (XML). W3C recommendation. World Wide Web Consortium. http://www.w3.org/XML/.

Charniak, E. (1991). Baysean Networks without Tears. AI Magazine, v. 12, n. 4, p. 50-63.

Costa, P. C. G. e Laskey, K. B. (2006). PR-OWL: A framework for probabilistic ontologies. In Proceedings of the 2006 conference on Formal Ontology in

Information Systems: Proceedings of the Fourth International Conference (FOIS 2006), pages 237-249. IOS Press, 2006. Available at http://portal.acm.org/citation.cfm?id=1566107.

Cuellar, A. A., Lloyd, C. M., Nielsen, P. F., Bullivant, D.P., Nickerson, D.P., Hunter, P.J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. Simulation, v. 79, n. 12, p. 740-747.

Devitt, A., Danev, B. e Matusikova, K. (2006). Constructing Bayesian Networks Automatically using Ontologies. In Proceedings of Second Workshop on Formal Ontologies Meets Industry (FOMI 2006).

Ding, Z. e Peng, Y. (2004). A Probabilistic Extension to The Web Ontology Language OWL. In Thirty Seventh Hawaii International Conference on System Sciences (HICSS 04), IEEE CS Press, 2004, pp. 40111.1.

Ding, Z., Peng, Y. e Pan, R. (2006). BayesOWL: Uncertainty modeling in semantic web ontologies. Soft Computing in Ontologies and Semantic Web, p. 3–29.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. International Journal of Human Computer Studies, v. 43, n. 5, p. 625–640.

Macedo, J. A. F. (2005). Um Modelo Conceitual para Biologia Molecular. PhD thesis, Departamento de Informática da PUC-Rio. Available at http://www.maxwell.lambda.ele.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=7939

Matos, E. E., Campos, F., Braga, R. e Palazzi, D. (2010). CelOWS: an ontology based framework for the provision of semantic web services related to biological models. Journal of Biomedical Informatics, v. 43, n. 1, p. 125-136.

Aplicações semânticas baseadas em microformatos

Vanderlei Freitas Junior[1], Daniel Fernando Anderle[1], Alexandre Leopoldo Gonçalves[2], Fernando Ostuni Gauthier[2], Denilson Sell[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia Catarinense
Santa Rosa do Sul, SC, Brasil.
[2]Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento
Universidade Federal de Santa Catarina
Florianópolis, SC, Brasil.

{junior, daniel}@ifc-sombrio.edu.br,
alexandre.gonçalves@ararangua.ufsc.br, gauthier@egc.ufsc.br,
denilson@stela.org.br

Resumo. *Os microformatos foram propostos com a finalidade de proporcionar semântica aos dados publicados na Web. Cerca de oito anos depois do seu surgimento e no aniversário de sete anos de fundação da comunidade Microformats.org, mantenedora do padrão, diversas aplicações foram e ainda são implementadas com vistas a fornecer semântica à internet, baseadas em microformatos. O presente trabalho propõe descrever alguns exemplos recentes de como esta tecnologia pode ser aplicadas no contexto da Web Semântica.*

Abstract. *Microformats have been introduced in order to provide semantics to web-published data. About eight years after their appearance and on the 7th anniversary of the creation of Microformats.org, a community that keeps up the standards, several applications were and continue to be implemented aiming to provide semantics to the internet based on microformats technology. The purpose of this paper is to describe some recent examples on how this technology can be applied in the context of Semantic Web.*

1. Introdução

Os microformatos foram apresentados no ano de 2004 como uma forma de agregar significado aos dados publicados na Internet. No ano seguinte, dadas as possibilidades de aplicações visualizadas por seus desenvolvedores, criou-se a comunidade Microformats.org, que vem ocupando-se por pesquisar e desenvolver novas especificações de microformatos, com vistas à proporcionar semântica às aplicações na Web.

Recentemente, em junho de 2012, a comunidade completou seus sete anos de existência, contando com um grande conjunto de microformatos classificados como estáveis, disponíveis para uso por toda a comunidade. Além disso, um grande volume de padrões em desenvolvimento, classificados como *drafts*, vêm recebendo cada vez mais atenção de diversos colaboradores ao redor do mundo.

Neste contexto, o presente trabalho propõe uma análise de algumas aplicações semânticas recentes baseadas em microformatos, demonstrando sua aplicabilidade e a atualidade de suas propostas.

Para melhor compreensão, o presente artigo está organizado da seguinte maneira: a seção 2 apresenta o conceito dos microformatos. A seção 3, por sua vez, apresenta e detalha as principais aplicações semânticas verificadas na atualidade, sendo seguida pela seção 4, que realiza as considerações finais e pela seção de referências.

2. Microformatos

Os microformatos surgiram, de acordo com Allsopp (2007), em 2004, na conferência South by Southwest (SxSW), com o lançamento do XHTML Friends Network (XFN).

Na época, o movimento dos blogs estava em franca expansão, e autores de todo o mundo passavam a anotar suas postagens com o objetivo de indicar suas relações com outros autores de blogs dos quais se inspiravam. O XFN foi então desenvolvido com o objetivo de garantir esta anotação semântica, estabelecendo estas ligações de forma mais padronizada, tornando-se muito popular entre os autores de blogs (Allsopp, 2007).

A tecnologia é mantida pela comunidade Microformats.org. Fundada em 25/06/2005, atualmente se constitui na maior referência mundial no assunto.

Os microformatos são conceituados de acordo com Microformats.Org (2012) como um conjunto simples de dados formatados abertos:

> Projetado primeiro para seres humanos e máquinas em segundo lugar, microformatos são um conjunto simples de dados formatados abertos, construídos sobre as normas existentes e amplamente adotadas. Em vez de jogar fora o que funciona hoje, microformatos pretendem resolver os problemas mais simples primeiro, adaptando-se os comportamentos atuais e padrões de uso (por exemplo, XHTML, blogs).

Com o intuito de ampliar a compreensão acerca do conceito de microformatos, a comunidade Microformats.org (2012) afirma que os microformatos são:

> - Uma maneira de pensar sobre os dados.
> - Princípios de concepção de formatos.
> - Adaptado a comportamentos atuais e padrões de uso.
> - Altamente correlacionada com XHTML semântico.
> [...]
> - Um conjunto de padrões de formatos de dados abertos e simples, em que muitos estão ativamente em desenvolvimento e em implementação visando atender adequadamente blog´s estruturados e a publicação de microconteúdo na web em geral.
> [...]

Microformats.org (2012) ainda afirma o que não são os microformatos:

> - Uma nova linguagem.
> - Infinitamente extensível e aberto.
> - Uma tentativa de fazer com que todos mudem seu comportamento e reescrevam suas ferramentas.
> - Uma abordagem totalmente nova que joga fora o que já funciona hoje.
> - Uma panaceia para todas as taxonomias, ontologias, e outras abstrações.
> - Definição abrangente demais ou considerada impossível de ser implementada.

O desenvolvimento de microformatos é baseado em um conjunto de princípios,

especificados pela comunidade Microformatos.Org (2012):

> - Resolver um problema específico.
> - Iniciar o mais simples possível.
> - Projetado para os seres humanos em primeiro lugar, para as máquinas de segundo.
> - Reutilizar blocos de construção de padrões amplamente adotados.
> - Modularidade / incorporabilidade.
> - Permitir e incentivar desenvolvimentos descentralizados de conteúdos e serviços.

Percebe-se, de acordo com Mrissa, Al-Jaba e Thiran (2008), que os microformatos oferecem como benefício a possibilidade de análise automática de informação na web. Por outro lado, permitem também a exportação de informação padronizada para aplicações externas. Oferecem ainda a possibilidade do seu uso por humanos através de *plugins* disponíveis para os navegadores atuais, permitindo, por exemplo, que a informação de um evento disponível em um site da Web no padrão de microformatos seja importada automaticamente, fazendo com que um novo compromisso seja criado na agenda do usuário com os dados do evento.

Khare (2006) e Stolley (2009) acrescentam que microformatos são uma nova abordagem para a codificação de informação semiestruturada usando XHTML, permitindo a descrição de pessoas, lugares, eventos e outros tipos comuns de informações de forma legível aos humanos.

Os princípios conceituais e filosóficos de implementação dos microformatos fazem deles soluções relativamente simples, para a solução de problemas pontuais, agregando semântica aos dados disponíveis na Web e permitindo a integração destes dados com aplicações desktop. Estas características poderão ser verificadas em algumas das aplicações descritas na seção seguinte.

3. Aplicações semânticas

3.1. Busca por receitas do Google®

Uma implementação de microformatos ainda em fase inicial chamada hRecipe foi objeto de pesquisas e base para o lançamento de uma nova ferramenta de busca da Google®: Recipe View. O microformato hRecipe contém em sua especificação diversas tags para a anotação semântica de receitas culinárias. Esta proposta motivou a companhia a implementar uma ferramenta de busca específica que seja capaz de procurar indicações semânticas nas páginas publicadas na Web relacionada às receitas culinárias e apresentá-las em um resultado de busca específico.

A nova busca permite a localização de receitas a partir de um de seus ingredientes, fornecendo dados relativos ao tempo de preparo e avaliações de usuários. Uma busca comum, pelo nome de um ingrediente, poderia retornar um conjunto infinito de respostas, entretanto com um nível baixo de efetividade. Com a nova tecnologia, o usuário pode aplicar filtros específicos e localizar a receita desejada (Figura 1).

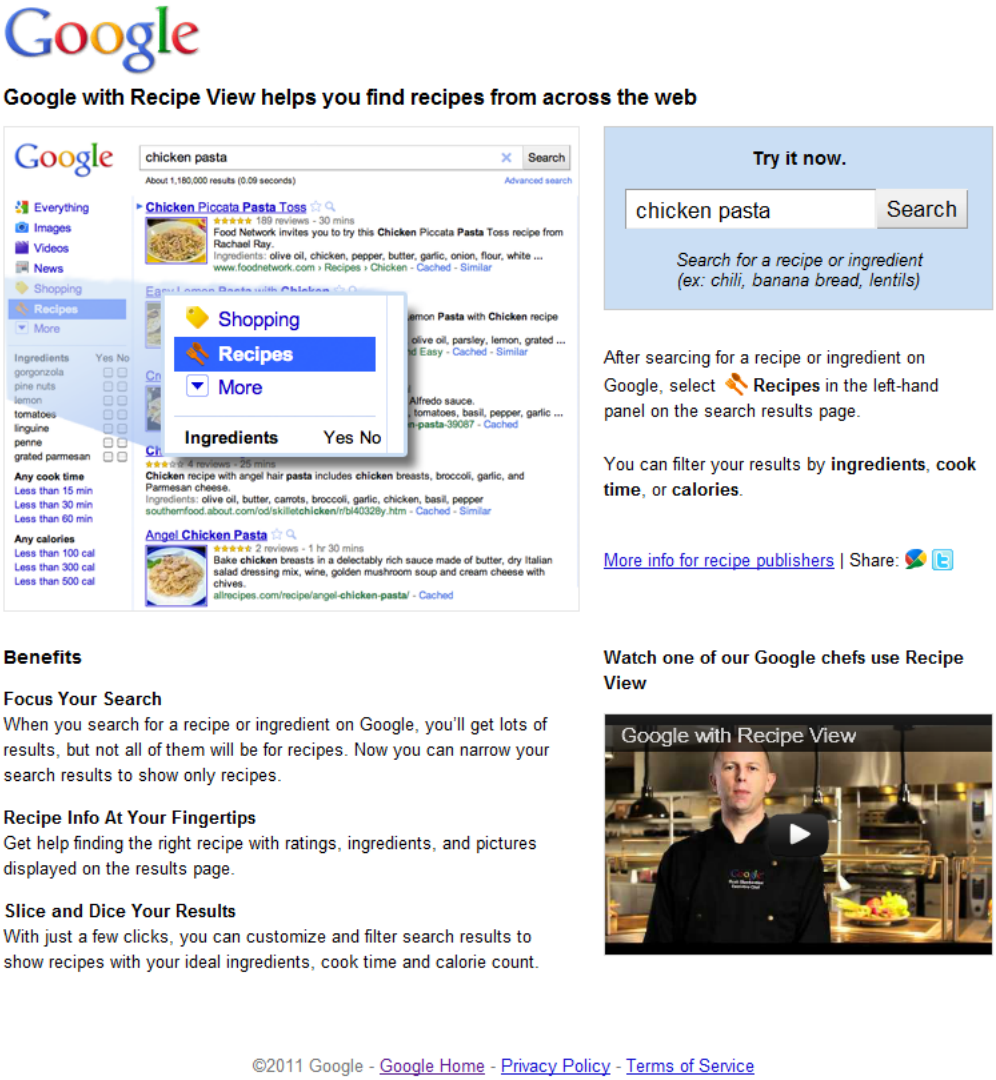


Figura 1 – Google® Recipe View

3.2. hCard e hCalendar no Facebook®

A rede social Facebook® passou a realizar a marcação de eventos cadastrados por seus usuários através do microformato hCalendar. Esta solução permite que os eventos cadastrados pelos usuários sejam indexados mais facilmente pelas ferramentas de busca e seus dados manipulados de forma mais complexa, possibilitando novos arranjos de informações (Microformats.org, 2012).

Além dos dados dos eventos, a rede social também faz uso do microformato hCard, permitindo a identificação dos dados de seus usuários de forma semântica.

3.3. Rich Snippets

Outra aplicação semântica baseada em microformatos, implementada pela Google®, são as chamadas Rich Snippets. Snippets são informações adicionais apresentadas nos resultados de busca da Google, obtidos através da análise semântica do conteúdo das páginas publicadas na Web (Google, 2012).

Esta tecnologia permite a visualização de dados contextualizados acerca da pesquisa realizada, juntamente dos resultados encontrados, fazendo com que o usuário possa decidir a relevância do site recuperado em relação à sua necessidade de busca.

De acordo com Google (2012), são também permitidas as anotações semânticas utilizando-se as tecnologias de Microdados e RDFa.

A Figura 2 demonstra uma busca realizada na ferramenta de pesquisa Google®, com as palavras chaves "Torta", "de" e "Camarão". Ao recuperar os resultados, a ferramenta analisa os dados à procura de marcações semânticas e, ao encontrá-las, apresenta-as na forma de dados extras, neste caso demonstrado pelo número de resenhas, pelo tempo de preparo e pela avaliação dos usuários.

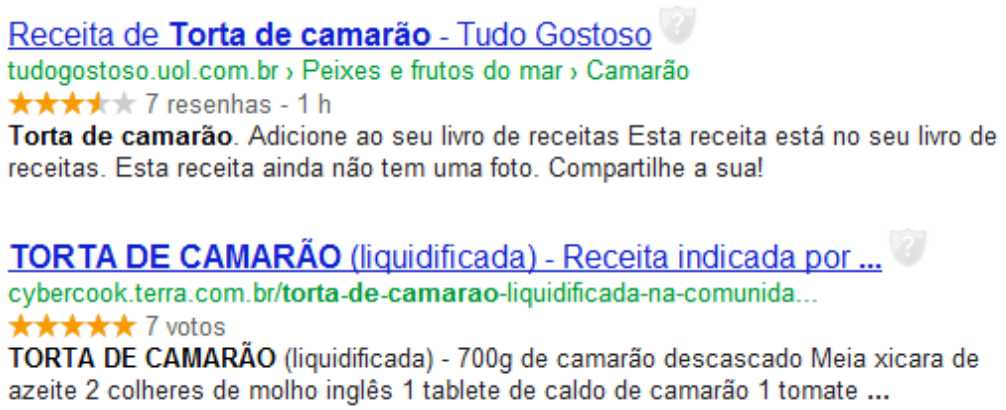


Figura 2 – Rich Snippets Google

3.4. Uso de vCard, hCard e QR Code no Moodle

O Moodle é um dos principais ambientes virtuais de aprendizagem disponíveis, contando com interfaces de relacionamento entre tutores e estudantes e de apoio à processos de aprendizagem.

Como forma de otimizar a publicação de dados pessoais de tutores e estudantes nesta plataforma, Dragolesco, Bucos e Mocofan (2011) propuseram uma metodologia que utiliza-se do microformato hCard para a veiculação destes dados, permitindo sua recuperação de forma mais facilitada.

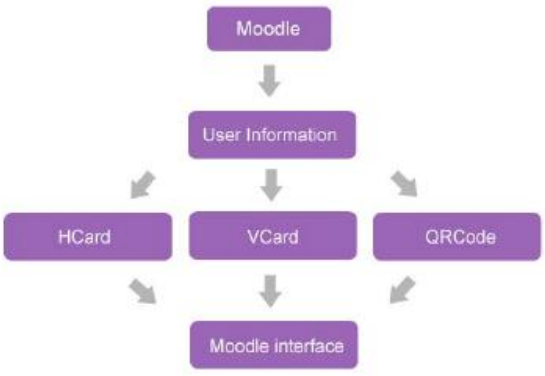


Figura 3 – Esquema proposto por Dragolesco, Bucos e Mocofan (2011)

A metodologia proposta consiste de um bloco Moodle que extrai as informações dos usuários da base de dados do sistema, processa-as e oferece três formas de apresentá-las na plataforma: hCard, vCard e QR Code (Figura 3).

Como trabalhos futuros, os autores propõem o uso do microformato hCalendar para a divulgação de eventos e compromissos na plataforma Moodle, além de localização mediante o uso do respectivo padrão.

4. Considerações finais

Os microformatos são uma das soluções possíveis para a anotação semântica dos dados distribuídos na Web. Neste sentido, o presente trabalho apresentou as bases teóricas, princípios e a filosofia dos microformatos, com ênfase à sua capacidade de enriquecimento semântico dos dados publicados na Web, demonstrando suas principais especificações e aplicações.

O desafio para a transformação dos dados publicados na web, proporcionando-lhes significado, semântica, ainda precisa ser enfrentado. Entretanto, a tecnologia de microformatos torna-se importante na medida em que proporciona uma solução relativamente simples para a identificação destes dados, permitindo que as máquinas possam processá-los de forma transparente em aplicações pontuais e específicas.

Como trabalhos futuros, propõe-se a identificação de aplicações baseadas em microformatos em ferramentas desktop, além do acompanhamento das aplicações baseadas nos microformatos em desenvolvimento.

5. Referências

Allsopp, John. (2007) "Microformats: Empowering your markup for web 2.0", Berkeley, CA, EUA: Friendsof.

Dragulesco, B.; Bucos, M.; Mocofan, M. (2011) "Using hCard and vCard for improving usability in Moodle", 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romenia, p. 473-476, mai.

Google. (2012) "Ferramentas Google para WebMasters", http://support. google.com/webmasters/bin/answer.py?hl=pt-BR&answer=99170. Jul.

Khare, R. (2006) "Microformats: The Next (Small) Thing on the Semantic Web?", IEEE Internet Computing, Standards, vol. 10 no. 1, p. 68-75, Jan/Fev.

Microformats.org. (2012) "About", www.microformats.org/about, Jul.

Mrissa, M.; Al-Jabar, M; Thiran, F. (2008) "Using Microformats to Personalise Web Experience", ICWE 2008 Workshops, 7th Int. Workshop on Web-Oriented Software Technologies – IWWOST, New York, USA. p. 63-68, Jul.

Stolley, K. (2009) "Using Microformats: Gateway to the Semantic Web – Tutorial", IEEE Transactions on Professional Communication, Vol. 52, nº3, p.291-302, Set.

Using ontologies to build a database to obtain strategic information in decision making

Érica F. Souza[1], **Leandro E. Oliveira**[1], **Ricardo A. Falbo**[2], **N. L. Vijaykumar**[1]

[1]Computação Aplicada – Instituto Nacional de Pesquisas Espaciais (INPE)
São José dos Campos – São Paulo – SP – Brazil

[2]Departamento de Informática – Universidade Federal do Espírito Santo – UFES
Vitória – Espirito Santo – ES – Brazil

{erica.souza, vijay}@lac.inpe.br, leandro.oliveira5@fatec.sp.gov.br,

falbo@inf.ufes.br

Abstract. *The manipulation of ontologies in databases can represent gains in the recovery of strategic information in decision-making process within the software development organizations. The software testing processes are strategic elements to develop projects and to the quality of the final product. Thus, this study investigates strategies to promote data handling of testing processes that are generated from a testing ontology. For this, a knowledge database is structured in a dimensional model for Data Warehouse to support storage and processing of data to obtain strategic information that can facilitate decision making.*

Resumo. *A manipulação de ontologias em bancos de dados pode representar ganhos na recuperação de informações estrátegicas no processo de tomada de decisão dentro das organizações de desenvolvimento de software. Os processos de teste de software são elementos estratégicos para a condução de projetos de desenvolvimento e qualidade do produto final. Diante disso, este trabalho tem como objetivo investigar estratégias que possam promover a manipulação de dados de processos de teste que são gerados a partir de uma ontologia de teste de software. Para isso, estrutura-se uma base de conhecimento em um modelo dimensional de Data Warehouse que apoie o armazenamento e o processamento dos dados para obtenção de informações estratégicas que podem facilitar a tomada de decisão.*

1. Introduction

With the exponential growth of data from several different sources of knowledge within an organization, it becomes necessary to provide automatized support for tasks of acquiring, processing, analyzing and disseminating knowledge. Organizations need to effectively manage the information generated in its production environment to promote the improvement of the processes used to generate knowledge and also support future decisions. Such data can provide important information for decision making, involving the identification and implementation of corrective actions.

One of the characteristics of software engineering projects is to deal with a great deal of information that are generated and manipulated. People involved in the project

face problems, such as: organize in a systematic way the information generated through the software process; reuse the knowledge generated from one project to another; loss of intellectual capital of the organization due to better opportunities; and no knowledge representation [Andrade et al. 2010].

In the area of software development, testing is a critical factor in product quality, and thus there is a greater concern with related research. Studies indicate that the quality of the software product is strongly dependent on the quality of the processes that are part of the project, especially the software testing process. However, finding relevant information (knowledge) in these processes can be a difficult and complex task, and it is related mainly to the lack of semantics associated with the large volume of information. There is a need to represent knowledge, to make it affordable and manageable. In this context, ontologies have been pointed out as an important way for representing knowledge [Rios 2005].

The manipulation of big ontologies with a high number of instances in the form of text files has a number of disadvantages, such as processing and query optimization [Filho et al. 2010]. Because of this, ontologies can be incorporated into a knowledge base to facilitate its management and access. Depending on the structure the database is created, the analysis of large data volumes and mining strategic information can facilitate decision making. Related work is found in [Astrova et al. 2007], [Vysniauskas and Nemuraite 2006], but they propose approaches that transform ontology representation into a relational database, but not in a dimensional model as this enables dealing with large volumes of information.

Given the above context, this paper aims to investigate strategies that can promote the manipulation of data generated from large ontologies. For this, a knowledge base in a dimensional model for Data Warehouse is structured to support the storage and processing of data to obtain strategic information that can facilitate decision making. A software testing ontology is being developed to support this work and instances of this ontology is used as a data source for the structure created. Section 2 briefly discusses ontologies and storage structures. Section 3 presents the proposed structure to store ontology data. Section 4 presents conclusions and future directions to follow.

2. Ontologies and structures for storage

During the last decades, ontologies have been shown useful in the field of Computer Science [Guizzardi 2005]. In a nutshell, an ontology is a formal specification of a shared conceptualization [Gruber 1993], i.e., a description of concepts and relationships that may exist for an agent or an agents community. Representation of a shared conceptualization requires a representation language. There are many representation languages. Some are defined based on the syntax of the eXtensible Markup Language (XML), like Resource Description Framework (RDF), Ontology Interchange Language (OIL) and Web Ontology Language (OWL). There is also graphical language for ontologies; an example is the Graphical Language for Expression Ontologies (LINGO) [Falbo et al. 1998].

Literature provides several tools to store and manipulate content from an ontology in a database and several strategies can be found [Filho et al. 2010]. Most tools use relational databases. Depending on the number of instances of an ontology, it becomes necessary to create structures that support high volumes of data and allow employing

techniques to find relationships among these data. An alternative is the use of Data Warehouse (DW) as the storage structure. It is a large repository of integrated data obtained from several sources for the specific purpose of data analysis [Christian et al. 2010]. A DW may take on different models: Cube and Star. The difference is in the Database Management System (DBMS). When the dimensional model is implemented in a relational database it is implemented as a Star architecture and when implemented in a multidimensional database it is known as Cube. Considering that DW stores a large volume of data, it optimizes and reduces the complexity of consultations, thus decreasing the response time and gaining in performance.

3. Proposed structure for storage of ontology content

The ontology used in this paper is in the context of the Ph.D. thesis of the first author [Souza 2011]. The ontology aims at software testing for Knowledge Management (KM). Test activity is incorporated into a process as it consists of several steps to add quality to the final products [Bastos et al. 2007]. Since the test activity is a process, the improvement can be incorporated with the use of KM and ontologies are identified today as being crucial in the KM in processes improvement.

Given the complexity of the software testing domain, an ontology and its sub-ontologies were created and used. Currently the main ontology created has: Steps, Techniques, Types, Artifacts and Environment. SABiO (Systematic Approach for Building Ontologies) was adopted to develop the software testing ontology [Falbo 2004].

As the ontology is still under development it may suffer some changes. However, though in the initial phase, it is already capable of describing execution phase of the tests, and from this premise that DW will be created to store ontology data. This phase contains data related to dates, for example, date of test case execution, date of defect submission and date of defect correction.

Instances of the software test ontology were extracted from an actual project developed at the (*Technological Institute of Aeronautics - ITA*) - (*Project of Amazon Integration and Cooperation for Modernization of Hydrological Monitoring - ICA-MMH B*) [Cunha 2010]. We used test data generated from Organizational Testing Management Maturity Model (OTM3) testing process [Lamas et al. 2010].

For converting the content of an ontology written in RDF into a DW architecture, we follow the process shown in Figure 1.

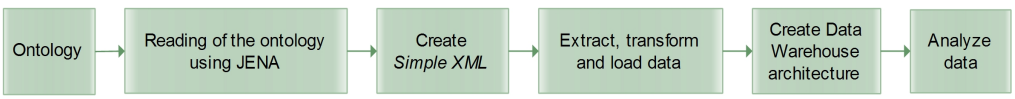


Figure 1. Process Proposed for conversion of an ontology in a DW dimension

To enable reading the ontology in the Pentaho Data Integration tool it was necessary to transform the RDF/OWL in a simple XML and export the data of the instances for the DW model created. For this, we used the Jena framework [Jena 2012] that provides an Application Programming Interface (API) in Java allowing writing, reading and extracting the description of fields, classes and instances from a file in the RDF/OWL in XML format. With the support of Jena framework, a Java class that converts the data into a pure

XML was developed, that is, using only the native tags of XML without the RDF/OWL specific tags. We call this *simple XML*, as shown in step three in Figure 1. The XML with simple pattern is to be read by Pentaho.

From the *simple XML* Pentaho is used to extract, transform and load the data for DW using ETL (Extract Transform Load). The data is loaded in DW table of facts. Star model was chosen to create the DW. The model consists of table *ft_test_execution* which is the table of facts, and five dimensions, namely: (i) three dimensions of time: *lk_execution_time* contains the date of the test execution, *lk_submission_time* refers to the date of a defect submission, and *lk_last_change_time* refers to the date of the last change made in the request to repair a bug; the *lk_team* dimension contains information about the tester, such as the name of the tester and his or her level within the team; and *ii*) the *lk_project* dimension contains information with respect to the project for which the test was performed. Figure 2 shows the Star model created with its respective tables.

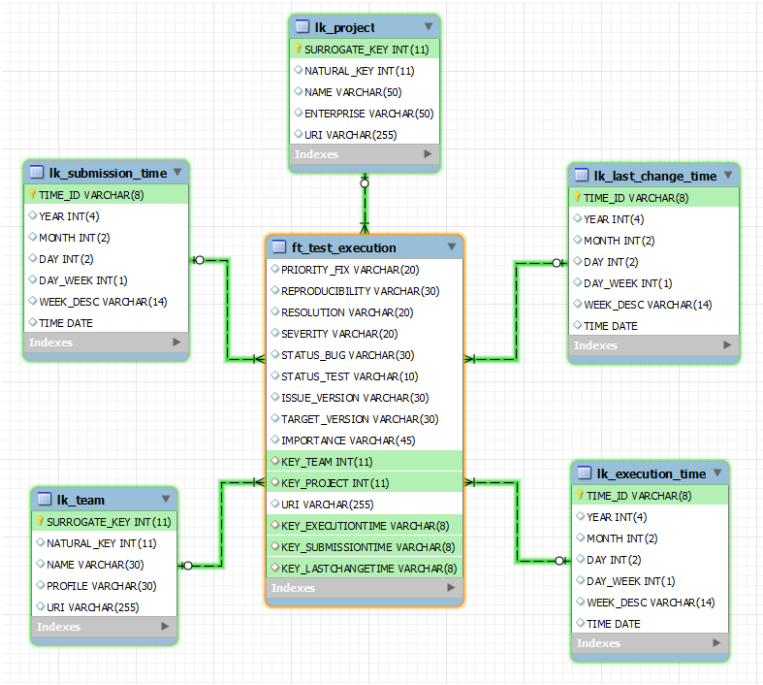


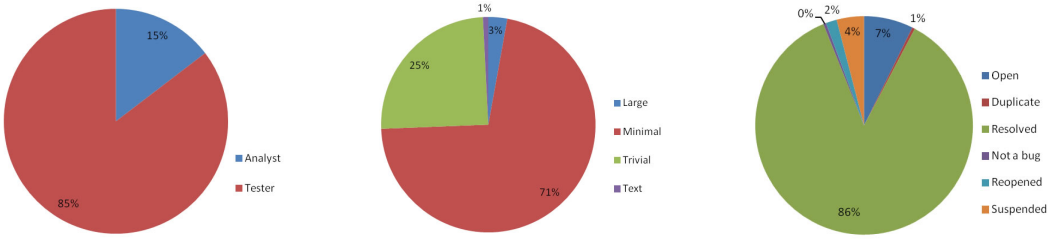
Figure 2. Development of Star Model

In order to see the contribution of the methodology, some questions (that might be important for decision makers) were posed. The idea is whether the table of facts can, in fact, answer such questions in a satisfied manner and useful to decision makers or managers. Therefore, just as an example, the following questions were defined to exercise the table of facts and the different dimensions of the DW:

1. What is the average time between the report of a defect and the test run to check if the bug was repaired? To this question the result obtained was an average of 5.85 days. This information can be useful for the manager or responsible for project to analyze whether the time between a request and the execution of the tests are on schedule.
2. What is the percentage of tests executed per profile of the team? Figure 3(a) shows the results obtained for this question. Most tests were executed by team members

with the position of tester, which is logical since the rest of the tests were executed by analysts and the main function of the analyst is to develop test plans and not to execute them. The result is consistent with the reality of a team of software testing.

3. How severe are the defects reported? Most defects reported have minimal severity and only 3% of the defects are of large severity. This suggests that the process of system development is a reasonable quality control (Figure 3(b)).
4. What is the relation between testing and defect solving? Most of the reported defects have been solved so far. Only 2% of the requests were reopened indicating a recurrence of an already reported defect, and only 1% of duplicate requests that are still under correction (Figure 3(c)).



(a) Number of Executions by profile.

(b) Number of tests by defect severity.

(c) Number of tests by resolution of the defect.

Figure 3. ***Results of preliminary analyzes***

4. Conclusions

This paper presented structuring and storing of ontology content in a DW model. We used a preliminary ontology of software testing process that is under development. The data analyzed refers to the test execution phase. The DW model created is expandable and may include other phases within the process of software testing.

The model facilitates queries and can also provide strategic information that can be used to improve the development process as well as for the process of software testing. The model can be enriched and provide more information that can be used in decision making. Some difficulties were encountered as lack of support for the Jena framework and real case studies with test procedures clearly defined.

Future directions include the automation of data entry in the ontology from other sources, model expansion to store data from other phases of the software testing process and integration with the software development process.

Acknowledgements

FAPESP and CNPq (PIBIC) for the financial support. ITA, Brazilian Water Agency (ANA), Brazilian Agency of Research and Projects Financing (FINEP) and the Casimiro Montenegro Filho Foundation (FCMF) for providing the data of Project FINEP 5206/06 for this work.

References

Andrade, M. T. T., Ferreira, C. V., and Pereira, H. B. B. (2010). Uma ontologia para a gestão do conhecimento no processo de desenvolvimento de produto. *Gestão e Produção*, 17:537–551. http://dx.doi.org/10.1590/S0104-530X2010000300008. Access in: Ago 2012.

Astrova, I., Korda, N., and Kalja, A. (2007). Rule-based transformation of sql relational databases to owl ontologies. *In: Proceedings of the 2nd International Conference on Metadata & Semantics Research*.

Bastos, A., Rios, E., Cristalli, R., and Moreira, T. (2007). *Base de conhecimento em testes de software*. Martins Editora Livraria, São Paulo, 2 edition.

Christian, S. J., Pedersen, T. B., and Thomsen, C. (2010). *Multidimensional Databases and Data Warehousing*, volume 2. Morgan and Claypool, 1 edition.

Cunha, A. M. (2010). Relatório Técnico do 5° Semestre do Projeto FINEP 5206/06. Technical report, São José dos Campos.

Falbo, R. A. (2004). Experiences in using a method for building domain ontologies. *In: International Workshop on Ontology in Action*, pages 474–477. Banff, Canada.

Falbo, R. A., Menezes, C., and Rocha, A. (1998). A systematic approach for building ontologies. *In: In Proceedings of the 6th Ibero-American Conference on AI, IBERAMIA98*. Lisbon, Portugal.

Filho, S. N. V., Moura, A. M. C., and Cavalcanti, M. C. R. (2010). Armazenamento e manipulação de ontologias utilizando sistemas gerenciadores de banco de dados. Technical report, Instituto Militar de Engenharia (IME), Rio de Janeiro.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *In: Formal Ontology in Conceptual Analysis and Knowledge Representation*. Padova, Italy.

Guizzardi, G. (2005). Ontological foundations for structural conceptual models. *Telematica Institute Fundamental Research Series, The Netherlands*. ISBN 90-75176-81-3.

Jena (2012). Apache Software Foundation - Jena. http://jena.apache.org/documentation. Access in: Aug. 2012.

Lamas, E., Souza, E. F., Nascimento, M. R., Dias, L. A. V., and Silveira, F. F. (2010). Organizational testing management maturity model for a software product line. *In: Seventh International Conference on Information Technology, ITNG'2010*, pages 1026–1031. Las Vegas, Nevada, USA.

Rios, J. A. (2005). Ontologias: alternativa para a representação do conhecimento explícito organizacional. *In: Proceedings CINFORM - Encontro Nacional de Ciência da Informação VI*. Salvador, Bahia.

Souza, E. F. (2011). Estratégias de reuso para melhoria de processo de teste de software baseado em ontologias. Technical report, INPE, São José dos Campos/SP. http://urlib.net/8JMKD3MGP7W/3BFFA9H. Access in: June 2012.

Vysniauskas, E. and Nemuraite, L. (2006). Transforming ontology representation from owl to relational database. *In: Information Technology and Control*.

A Semantic web approach for e-learning platforms

Miguel B. Alves[1]

[1]Laboratório de Sistemas de Informação, ESTG-IPVC
4900-348 Viana do Castelo.

mba@estg.ipvc.pt

Abstract. *When lecturers publish contents in an e-learning platform like a course degree or the bibliography for a given course, normally they do that either by uploading a document (MSWord, pdf) or by creating a resource that will be part of a webpage. In both cases, this kind of information is static and is only useful for human readers, the information is not open to other systems, and in general to the world. In this work, we propose to enrich an e-learning platform with semantic web content so that information is available to external systems. Concretely, we will develop our approach over the Moodle platform, the widest e-learning platform used. Moreover, we will focus on representing the course's degree and its bibliography.*

1. Introduction

Lets consider that a university wants to know which books are recommended in all the courses of its degrees. They need this information to know if its library is well-served of recommended books. The "well-served" conception has two dimensions: a) all the recommended books should exist in the library; b) all recommended books should exist in the library in the right quantity, which means, there are books that are widely used and there should be several copies available. The university policy forces teachers to publish the bibliography of each course in the e-learning platform. However, they concluded that all information is available to the users but it cannot be accessed in a structured way. It is not a desirable solution to develop another system for that particular purpose because of its cost. The ideal is to access the information that is published in e-learning platform, a web-based system, in a structured way. This relates immediately with the Semantic Web initiative where contents are machine interpretable. The Semantic Web is a proposal of the World Wide Web inventor Tim Berners-Lee and colleagues[Berners-Lee et al. 2001] that the Web as a whole can be made more intelligent and perhaps even intuitive about how to serve a user's needs. Berners-Lee observes that although search engines index much of the Web's content, they have little ability to select the pages that a user really wants or needs. He foresees a number of ways in which developers and authors, single or in collaboration, can use self-descriptions and other techniques so that context-understanding programs can selectively find what users want. However, despite the designation, Semantic Web is not only for web content but, in general, it is an interpretable machine approach and this has application in many areas. The global vision of the development of the Semantic Web is to make the contents of the Web machine interpretable. To achieve this overall goal, ontologies play an important role as they give the means for associating precisely defined semantics with the content that is provided by the web. Ontologies are defined as the representation of the semantics of terms and their relationships. They consist of concepts, concepts' atributes and relationships between concepts, all expressed in linguistic terms[Guarino et al. 1993].

This paper reports the project developed in a polytechnic institute which uses Moodle as web e-learning platform. The purpose of the project is the development of Moodle plug-ins to bring the bibliographic information out of the e-learning system. Moreover, the institute requests the teacher of a given course to do the mapping between course contents and the recommended bibliography. The purpose is to create the necessary infrastructure that allows the future development of a system where the student can consult the bibliography of a given subject in a course. He will be able to see if a book is available in library, connect to an on-line sales company if he desires to buy the book, and so on. This document is organized as follows. Section 2 describes our approach to enrich a e-learning platform with semantic information. Section 3 discusses future work, and the paper ends in Section 4 with conclusions.

2. Semantic Web Approach in E-Learning Systems

In [Diaconescu et al. 2008] are discussed the advantages of a semantic web approach in e-learning systems. Briefly, representing data in the Resource Description Framework (RDF) [Brickley and Guha 2004] [Tauberer 2006], instead of a traditional approach as relational databases, is a shift to the open world with many distributed resources, identified by URIs as a mechanism for referring to global entities on which there is some agreement among multiple data providers. Queries can be performed not only over a single database, but over the content of several distributed educational systems, including resources, which are externally available on the Web. RDF is a W3C [1] standard for modelling and sharing distributed knowledge based on a decentralized open-world assumption. RDF was designed as a metadata model and it has come to be used as a general method for conceptual description or modelling of information that is implemented in web resources. Knowledge is expressed by triples consisting of subject, predicate and object (like a short English sentence), also known as statement, forming RDF graphs. In the Web, we use RDF to make statements about resources. In particular, we can classify the resource. RDFa [W3C 2012] is a specification for attributes to express structured data in any markup language. RDFa, which means RDF in HTML *a*ttributes, adds a set of attribute-level extensions to XHTML for embedding rich metadata within Web documents. This allows web pages to be understandable by machines, give information to the browsers and search engines about the pages. A Semantic Web approach also allows reasoning over the contents. Rules can be defined over the RDF statemensts, and those rules can be extended any time. To describe a bibliography, we make use of the Bibliographic Ontology (BIBO)[D'Arcus and Giasson 2009], which is an ontology for the semantic Web to describe bibliographic things like books or magazines. To describe a bibliographic resource, BIBO makes use of the Dublin Core ontology. Dublin Core element set [DCMI 1998] is a flexible and usable metadata schema enabling information exchange and integration between digital sources. It is widely used by almost all digital libraries since it is simple, small and easily expandable, and provides qualifiers that enable the semantic expression. The significant role of DC in data exchange is obvious due to the fact that there are mappings from and to it by many widely used metadata schemas [Day 2002]. Dublin Core is widely used to describe digital materials such as video, sound, image, text and composite media like web pages. Although Dublin Core can be used to describe bibliography, we use BIBO Ontology because it has a higher degree of richness to describe

[1] `http://www.w3.org/`

books, for example, isbn10 and isbn13 properties. One the of purposes of the project is to encourage the teachers to detail the information mapping course contents with bibliography, driving the student exactly to the contents that he should focus. For that, we need semantic information about the course, its contents and a way to map the contents with bibliography. We make use of the Academic Institution Internal Structure Ontology (AIISO) [Styles and Shabir 2008], that provides classes and properties to describe the internal organizational structure of an academic institution and TEACH, the Teaching Core Vocabulary [Kauppinen and Trame 2011], which is a lightweight vocabulary providing terms to enable teachers to relate things in their courses together.

2.1. Linked Open Data Project

The aim of Linking Open Data Project [Bizer et al. 2009] is using the web to create typed links between data from different sources via mapping of ontologies. In this way, instead of having isolated islands we have global interlinked datasets. Linked Data refers to data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets and can be accessed by them. Linked Data principles provide a basic recipe for publishing and connecting data using the infrastructure of the Web while adhering to its architecture and standards.

All ontologies used in this work belong to Linked Open Data, under `LinkedUniversities.org`. *Linked Universities* is an alliance of european universities engaged into exposing their public data as linked data. Using technologies such as RDF and SPARQL, it gives direct access to information such as their publications, courses, educational material, etc.

2.2. Semantic Annotation

In this subsection we will detail how the webpages can be annotated to be enriched with semantic information. For a better understanding of this work, next we summarize the namespaces used. Besides, to save space, in all of the HTML excerpts below we omit *namespaces* declaration.

```
xmlns:dc='http://purl.org/dc/terms/'
xmlns:aiiso= 'http://purl.org/vocab/aiiso/schema#'
xmlns:teach='http://linkedscience.org/teach/ns#'
xmlns:bibo='http://purl.org/ontology/bibo/'
```

Lets consider the following excerpt of HTML from a web page course in Moodle.

```
<div typeof="bibo:Book"resource="3642159699≫
<h2><span property="dc:title≫ A developer's guide to the semantic web</span></h2> <p>
Author:  <span property="dc:creator≫Liyang Yu</span><br>
ISBN-10:  <span property="bibo:isbn10≫3642159699</span> | ISBN-13:  <span
property="bibo:isbn13≫978-3642159695</span> <br> </p> </div>
```

The purpose is to enrich the web page with semantic information that can be used outside of Moodle. The resulting page might be:

```
<div typeof="bibo:Book"resource="ISBN:3642159699≫
<h2><span property="dc:title≫ A developer's guide to the semantic web</span></h2>
<p> Author:  <span property="dc:creator≫Liyang Yu</span><br>
ISBN-10:  <span property="bibo:isbn10≫3642159699</span> | ISBN-13:  <span
property="bibo:isbn13≫978-3642159695</span><br> </p> </div>
```

As we can see, the web page was enriched with semantic information that can be used by other systems. BIBO ontology is used to describe the book. Next, we list the RDF information extracted from the previous example using a RDFa parser service [2].

```
<ISBN:3642159699> <rdf:type> <bibo:Book> .
<ISBN:3642159699> <dc:title> "A developer's guide to the semantic web".
<ISBN:3642159699> <dc:creator> "Liyang Yu".
<ISBN:3642159699> <bibo:isbn10> "3642159699".
<ISBN:3642159699> <bibo:isbn13> "978-3642159695".
```

Now, consider a course named *Semantic Web*. One of the topics of this course is *The RDF language and its XML serialization*. Next, we list an excerpt of a HTML webpage to describe the course and its contents, enriched with semantic information.

```
<div typeof="aiiso:Course"resource="semweb≫
<h2><span property="dc:title≫Semantic Web</span></h2>
<p> </div>
<div typeof="aiiso:Programme"resource="semweb-topic2≫
<span property="dc:title≫The RDF language and its XML serialization</span>
<span rel="aiiso:knowledgeGrouping"resource="urn:semweb≫</span> </div> </p>
```

The course and its contents are modelled with AIISO ontology. A course is of the type of *Course* class. We modelled the topics of the course with the class *Program*. Both *Course* and *Program* are sub-classes of the class *KnowledgeGrouping*, which represents a collection of resources, learning objectives, timetables, and other material. A *KnowledgeGrouping* may be contained by another *KnowledgeGrouping* or an organizational Unit using the *knowledgeGrouping* property. We use this property to relate a course with its contents. Additionally, this property allows the definition of sub-topics, in a hierarchical view.

Now, lets consider that one recommended reading to the topic *The RDF language and its XML serialization* of the *Semantic Web* course is the chapter 2 of book *A developer's guide to the semantic web*, named *The Building for the Semantic Web:RDF*. This mapping can be done in two ways: a) when the teacher is editing the course content, he indicates the recommended reading for a given topic; b) when the teacher is editing the bibliography, he indicates the topics which are supported by that book or chapter. To model chapters of a given book, BIBO ontology has the class *Chapter*. We make use of Dublin Core property `isPartOf` to define that a chapter belongs to a given book. The following HTML webpage excerpt shows the semantic information associated with a book chapter:

```
<div typeof="bibo:Chapter"resource="ISBN:3642159699-chapter2≫
Chapter 2 - <span property="dc:title≫ The Building for the Semantic
Web:RDF</span>
<span property="bibo:chapter≫2</span>
<div rel="dc:isPartOf"resource="ISBN:3642159699≫</div></div>
```

Let us consider first the model where the teacher associates the recommended reading when he is editing the course content. For that, we make use of the property `reading` of TECH ontology. The HTML excerpt below shows how modelling is done.

```
<div typeof="aiiso:Course"resource="semweb≫
<h2><span property="dc:title≫Semantic Web</span></h2>
```

[2] `http://rdf-in-html.appspot.com/`

```
<p> <ul> <li typeof="aiiso:Programme"resource="semweb-topic2≫
<span property="dc:title≫The RDF language and its XML serialization</span>
<span rel="aiiso:knowledgeGrouping"resource="urn:semweb≫</span>
<span rel="teach:reading"resource="ISBN:3642159699≫</span>
</li> </ul> </p> </div>
```

The corresponding RDF triples are:

```
<urn:semweb> <rdf:type> <aiiso:Course> .
<urn:semweb> <dc:title> "Semantic Web".
<urn:semweb-topic2> <rdf:type> <aiiso:Programme> .
<urn:semweb-topic2> <dc:title> "The RDF language and its XML serialization".
<urn:semweb-topic2> <aiiso:knowledgeGrouping> <urn:semweb> .
<urn:semweb-topic2> <teach:reading> <ISBN:3642159699> .
```

Consider now that the teacher indicates the book or the chapters of books that support a given topic when he is editing the bibliography. For that, he make use of the property `isReferencedBy` of Dublin Core Ontology.

```
<div typeof="bibo:Chapter"resource="ISBN:3642159699-chapter2≫
Chapter 2 - <span property="dc:title≫ The Building for the Semantic Web:RDF</span>
<span property="bibo:chapter≫2</span>
<span rel="dc:isPartOf"resource="ISBN:3642159699≫</span>
<span rel="dc:isReferencedBy"resource="semweb-topic2≫</span> </div>
```

The corresponding RDF triples are:

```
<ISBN:3642159699-chapter2> <dc:title> "The Building for the Semantic Web:RDF".
<ISBN:3642159699-chapter2> <dc:isPartOf> <ISBN:3642159699> .
<ISBN:3642159699-chapter2> <dc:isReferencedBy> <urn:semweb-topic2> .
```

3. Future Work

The next step is to develop plugins to Moodle to permit inserting semantic information in webpages without needing technical knowledge. The user does not need to have knowledge about semantic web and is not expected to fill in the webpages with semantic information. That semantic information should be inserted by user-friendly tools, incorporated in Moodle as plugins. These user-friendly tools also can use information from ontologies. For example, to help the user in bibliography editing, an ontology like RDF Book Mashup[Bizer et al. 2007] can be used, helping filling in of data fields.

Another important task that is not directly related with this work but it is important to other projects is extracting the semantic information from webpages and keeping it in a database in order to be used. Jena GRDDL (Gleaning Resource Descriptions from Dialects of Languages) Reader[3] can be used to extract RDF data from HTML pages. This information can be kept in any RDF triple database or even in a relational database (however, this last option can result in drawbacks in using semantic reasoning). In [Diaconescu et al. 2008] is introduced how we can deal with semantic information.

In the future, it should be interesting to extend the semantic information to other fields of education, towards a completly linked university, using approaches and ontologies present in literature for that purpose, which we introduce in this work.

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

4. Conclusion

In this work, we presented a semantic web approach in Moodle, the widest e-learning platform used. The purpose is to enrich web contents with semantic information, opening the contents to the open world. Adopting the linked open data principles, the information contained in webpages is available to outside systems, which can read and interpret the information without any kind of specifications or protocols. This work is an on-going project to make the information in one polytechnic institute accessible to other systems that can be developed in the future. However, as these systems are not planned yet, the purpose is make the information available in a standard and open way. This work focuses on modelling courses and their contents and bibliography, allowing mappings between them.

Referências

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284:28–37.

Bizer, C., Cyganiak, R., and Gauss, T. (2007). The rdf book mashup: from web apis to a web of data. In *3rd Workshop on Scripting for the Semantic Web, ESWC, Innsbruck, Austria, June*, volume 6, pages 1613–0073.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Brickley, D. and Guha, R. V. (2004). Rdf vocabulary description language 1.0: Rdf schema. *W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/*.

D'Arcus, B. and Giasson, F. (2009). Bibliographic ontology specification. http://bibliontology.com/specification.

Day, M. (2002). Mapping between metadata formats. *UKOLN UK Office for Library and Information Networking*.

DCMI (1998). Dublin core metadata element set, version 1.0: Reference description. http://dublincore.org/documents/1998/09/dces/.

Diaconescu, I.-M., Lukichev, S., and Giurca, A. (2008). Semantic web and rule reasoning inside of e-learning systems. In Badica, C. and Paprzycki, M., editors, *Advances in Intelligent and Distributed Computing*, volume 78 of *Studies in Computational Intelligence*, pages 251–256. Springer Berlin / Heidelberg.

Guarino, N., Poli, R., and Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing.

Kauppinen, T. and Trame, J. (2011). Teaching core vocabulary specification. http://linkedscience.org/teach/ns/.

Styles, R. and Shabir, N. (2008). Academic institution internal structure ontology (aiiso). http://vocab.org/aiiso/schema.

Tauberer, J. (2006). What is rdf. http://www.xml.com/pub/a/2001/01/24/rdf.html.

W3C (2012). Rdfa 1.1 primer. http://www.w3.org/TR/xhtml-rdfa-primer/.

Ontologias para descrição de recursos multimídia: uma proposta para o CPDOC-FGV

Daniela L. Silva[1,3], Renato R. Souza [2],

Fabrício M. Mendonça [3], Maurício B. Almeida [3]

[1] Departamento de Biblioteconomia – Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514 - Goiabeiras – 29.075-910 – Vitória – Brasil

[2] Escola de Matemática Aplicada – Fundação Getúlio Vargas
Praia de Botafogo, 190 – 22.250-900 – Rio de Janeiro – Brasil

[3] Escola de Ciência da Informação – Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 – Campus Pampulha – 31.270-901 – Belo Horizonte – Brasil

danielalucas@hotmail.com, renato.souza@fgv.br,
fabriciommendonca@gmail.com , mba@eci.ufmg.br

Abstract. *This paper describes a proposal for building an ontology in the multimedia description domain, in the context of the center for teaching and research in the Social Sciences and Contemporary History (CPDOC) from the FGV. It also presents the results from a state-of-art review study of the multimedia and controlled vocabularies available, and its relation with the Semantic Web Linked Data recommendation.*

Resumo. *O artigo descreve uma proposta para construção de uma ontologia para o domínio da descrição multimídia envolvendo o Centro de Pesquisa e Documentação de História Contemporânea do Brasil (CPDOC) da FGV. Apresenta-se também um resultado conciso do estado da arte da temática de vocabulários e metadados multimídia e sua relação com Linked Data.*

1. Introdução

O crescimento exponencial de informações, ocasionado principalmente pelas facilidades introduzidas pelas tecnologias da informação e comunicação, vem impondo desafios no processo de produção, organização e disseminação de informação em Unidades de Informação como Arquivos, Bibliotecas, Museus, Centros de Documentação e Projetos de Memória.

Pesquisas têm sido desenvolvidas progressivamente nos campos das Ciências da Computação e da Informação, visando a estudos sobre a problemática do excesso de informações e sua organização, com o objetivo de melhorar a eficácia dos sistemas de recuperação de informação. Podemos citar, dentre outras, algumas pesquisas nessa perspectiva voltadas à exploração semântica da informação, tais como: a) a Web Semântica e sua proposta emergente de *Linked Data* que intencionam criar

metodologias, tecnologias e padrões de metadados para aumentar o escopo da interoperabilidade e da integração plena de informações heterogêneas entre sistemas de informação [Berners-Lee, Hendler e Lassila 2001] [Berners-Lee 2006]; e b) instrumentos de representação de relacionamentos semânticos e conceituais como ontologias e vocabulários controlados [Gruber 1993], [Guarino 1998], [Silva, Souza e Almeida, 2008] objetivando endereçar problemas relacionados à interoperabilidade de sistemas e bases de dados, além das dificuldades intrínsecas à manipulação da linguagem natural como, por exemplo, as questões de polissemia e sinonímia.

Uma das principais mudanças que reflete a Web é a desterritorialização do documento e a sua desvinculação de uma forma física tradicional como o papel, possibilitando uma integração entre diferentes suportes (texto, imagem, som, vídeo) e a modificação na forma linear de acesso promovida pela inserção das tecnologias hipertexto e hipermídia. Em esfera global, observam-se nos últimos três anos [Schandl et al. 2011] um crescimento significativo de dados semanticamente relacionados e distribuídos na Web – o que se tem denominado na literatura de *Linked Data*. Nesse contexto, padrões de metadados recomendados pelo *World Wide Web Consortium* (W3C) vêm sendo utilizados para descrever e representar recursos multimídia, possibilitando ampliar os pontos de acesso e melhorar a gestão, a organização e a recuperação de acervos digitais. Entretanto, o relacionamento entre multimídia e *Linked Data* ainda é pouco estudado nas comunidades multimídia e ciência da Web [Schandl et al. 2011], abrindo-se oportunidades de pesquisa voltadas a tecnologias eficientes para geração, exposição, descobrimento e consumo de recursos multimídia semanticamente vinculados na Web.

Este artigo objetiva apresentar uma proposta endereçada à construção de uma ontologia de domínio da descrição multimídia para o Centro de Pesquisa e Documentação de História Contemporânea do Brasil (CPDOC). O Centro é dedicado ao estudo e à preservação da memória do país e, atualmente, abriga o mais importante acervo de arquivos pessoais de homens públicos no Brasil (em manuscritos, impressos, fotografias, áudios e vídeos) organizado em sistemas de informações com características próprias. A ontologia de domínio proposta busca a melhoria dos processos de organização da informação do acervo multimídia do CPDOC e a integração de seus sistemas junto a Web de dados.

O presente artigo está estruturado da seguinte forma: na seção 2 são apresentados conceitos, tecnologias e problemas que circundam a temática *vocabulários e metadados multimídia* e seu relacionamento emergente com o paradigma *Linked Data*; na seção 3 é descrita a metodologia para construção de modelos semânticos a um Centro de Pesquisa e Documentação; na seção 4 apresenta-se um resultado parcial de pesquisa sobre vocabulários considerados úteis para o contexto multimídia na Web, e que podem servir para reuso e extensão em um processo de construção de ontologias; e finalmente, a seção 5 é dedicada às considerações finais.

2. Descrição de recursos multimídia e Linked Data

Utilizar metadados é a forma mais comumente empregada para agregar semântica a informações [Gilliland 2000] com o propósito de facilitar a busca de recursos informacionais. No caso de recursos multimídia, os metadados podem ser usados tanto

para descrever atributos técnicos de baixo nível do conteúdo (cores, texturas, timbres de som, descrição de melodia) quanto para descrever características semânticas de alto nível como, por exemplo, classificação de gênero ou representação de informação sobre pessoas retratadas na mídia.

No escopo da Web Semântica [Berners-Lee, Hendler e Lassila 2001], os metadados são agregados através das chamadas linguagens de marcação (do inglês, *markup languages*). Estas linguagens, cujo padrão mais conhecido e utilizado é o XML (*eXtensible Markup Language*), definem *tags* ou marcações que são adicionadas aos dados a fim de indicar alguma informação importante. Ainda que o padrão XML tenha se tornado bastante popular, logo se percebeu que somente esse padrão não é suficiente para permitir a correta interpretação das informações por um sistema informatizado, pois tal sistema não consegue inferir, através das marcações, o que uma informação significa. Tal limitação pode acarretar deficiências nas buscas e na interoperabilidade entre sistemas.

Alternativas estão sendo propostas para este problema pelo W3C no projeto da Web Semântica. Uma dessas alternativas é a adoção do conceito de ontologias para a compatibilização de conceitos encontrados em bancos de dados dos mais diversos tipos na Web. As ontologias apresentam-se como possibilidades de representação de conhecimento em sistemas de informação na medida em que buscam organizar e padronizar conceitos, termos e definições aceitas por uma comunidade particular. Várias linguagens baseadas em XML têm sido propostas para representar ontologias como RDF (*Resource Description Framework*), RDF *Schema* e OWL (*Ontology Web Language*); além da linguagem de consulta para dados modelados em RDF, a SPARQL [Allemang e Hendler 2008].

O enriquecimento semântico sobre dados abertos e vinculados, também conhecido como iniciativa LOD - *linked open data* [Berners-Lee 2006], é uma abordagem recente proposta pelo W3C. A proposta é usar os padrões abertos concebidos pelo W3C em projetos para a Web Semântica a fim de interligar e anotar dados reutilizando vocabulários, ontologias e esquemas de metadados. Nesse sentido, busca-se uma visão integrada de dados e uma maximização da interoperabilidade semântica entre conjuntos de dados (*data sets*) de produtores e consumidores de conteúdo na Web. Os conjuntos de dados Geonames[1] e DBpedia[2] são comumente usados e fazem parte da "nuvem LOD[3]". Entretanto, seus esquemas (além de outros disponíveis na nuvem) não são suficientes para uma atribuição semântica satisfatória aos dados, pois não compreendem um modelo conceitual adequado para representar parte de suas realidades. Além disso, possuem deficiências na qualidade das informações publicadas na nuvem: i) falta de descrição conceitual nos conjuntos de dados; ii) ausência de *links* nos esquemas de dados; e iii) falta de expressividade semântica na representação de dados [Jain et al. 2010].

Provedores de conteúdo multimídia podem enriquecer semanticamente seus esquemas de metadados com especificações estruturadas e bem definidas de

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

[2] http://dbpedia.org/About

[3] Representação gráfica de fontes de dados populares e das ligações entre as mesmas. Cf. http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics

conhecimento (por meio de ontologias, por exemplo), viabilizando o consumo e o reuso de informações de alta qualidade e, muitas vezes, multilíngue fornecidas por bases de conhecimento publicamente acessíveis, como o DBpedia, por exemplo. Além disso, podem introduzir *links* para seus descritores de metadados aumentando a visibilidade e a expansão na cobertura de seus conteúdos na Web. Observam-se, assim, mudanças significativas nos modelos de organização e representação do conhecimento no espaço digital no que tange a propostas de melhorar os sistemas de busca e navegação por meio da agregação de abordagens semânticas aos recursos na Web, de forma a obter resultados mais significativos pelos usuários.

3. Metodologia para construção de modelos semânticos para o CPDOC integrados a Web de dados

Uma parte significativa dos conjuntos documentais do CPDOC encontra-se em formato digital e disponível para consulta online. Apesar de poderem ser acessados através do mesmo portal, possuem interfaces e processos de descrição e publicização distintos. São cerca de: 1,2 milhão de documentos manuscritos e impressos (ou 5.1 milhões de páginas); 80 mil fotografias; 6 mil horas de entrevistas em áudio e vídeo; e 8 mil verbetes de natureza biográfica e temática.

O CPDOC conta hoje com um projeto de integração de dados de seus sistemas de informação visando à criação de um portal semântico com interface única para buscas temáticas transversais e integradas. Foi engendrado para promover uma maior integração das bases de dados internas com as externas, como a própria Wikipédia, com benefícios no sentido de aumento da publicização e estruturação de redes sociais de colaboração para contribuições e eventuais correções para o acervo. O projeto prevê a criação de ontologias para descrição de recursos multimídia (áudio, vídeo, imagem, texto) e ontologias no domínio de história contemporânea.

No que tange à ontologia para o domínio da descrição multimídia, foco desta proposta de trabalho, o propósito é conceber modelos conceituais ontologicamente consistentes e bem fundamentados, isto é, dando ênfase à explicitação na semântica dos esquemas de dados internos e externos de interesse do CPDOC. Uma ontologia de domínio bem fundamentada é um modelo de domínio específico que se articula com um domínio de sistema de categorias formal e independente, denominado ontologias de fundamentação [Guizzardi e Wagner 2009]. As categorias ontológicas podem ser úteis no sentindo de esclarecer o significado pretendido dos termos adotados por meio de um conjunto de distinções semânticas, evitando ambiguidade e melhorando, principalmente, a qualidade na representação de dados no contexto *Linked Data*.

A proposta é construir a ontologia de domínio da descrição multimídia orientada por uma ontologia de fundamentação como, por exemplo, a *Unified Foundational Ontology* (UFO) [Guizzardi e Wagner 2009] e a *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) [Masolo et al., 2003], observando-se, ainda, vocabulários e metadados multimídia disponíveis na Web com vistas a reuso ou a extensão. A ontologia de domínio bem fundamentada para descrição multimídia será útil para a integração semântica entre as bases de dados do CPDOC e estas, por sua vez, ligadas com conjuntos de dados pertencentes à Web de dados. Tal integração será estabelecida por meio de modelos conceituais dos conjuntos de dados envolvidos

ligados à implementação da ontologia de domínio. Acredita-se que a utilização de um nível conceitual é relevante no sentido de abstrair características tecnológicas, além de fornecer uma descrição conceitual para conjuntos de dados e melhorar a compreensão humana e a atribuição semântica às máquinas.

4. Resultados parciais

O objetivo da presente seção é apresentar e descrever de modo sucinto alguns vocabulários (incluindo ontologias) que foram desenvolvidos nos últimos anos pelas comunidades de Web Semântica e *Linked Data*, os quais se mostram relevantes no contexto de marcação semântica para conteúdos multimídia. Tais vocabulários são considerados uma boa prática para reuso ou extensão [Schandl et al. 2011]. O Quadro 1 exibe os vocabulários. Para a exploração da literatura sobre vocabulários e metadados multimídia utilizou-se da técnica de pesquisa bibliográfica e documental em artigos científicos, livros e relatórios técnicos de pesquisa. Para a identificação de documentos relacionados à temática, foram consultadas bases de dados de documentos científicos no portal de periódicos da Capes e na biblioteca digital *Citeseer*. No que diz respeito ao portal de periódicos da Capes, as editoras consultadas foram: i) *Association Computing Machinery*; ii) *Journal Multimedia Tools and Applications;* e iii) *IEEE MultiMedia*.

Quadro 1: Vocabulários relevantes para o contexto multimídia

Vocabulário	Característica
Dublin Core	Fornece propriedades para descrever artefatos criados pelo homem como proveniência, formato, idioma, direitos autorais. Voltado ao domínio de metadados bibliográfico.
Friend of a Friend	Descreve pessoas, organizações e relacionamentos entre eles.
Basic Geo Vocabulary	Define propriedades para a representação de coordenadas geográficas (latitude, longitude e altitude).
Creative Commons	Fornece termos e classes para representar informação legal sobre obras, licenças associadas e permissão de distribuição e uso.
Review Vocabulary	Fornece termos que representam revisões, críticas e comentários para objetos arbitrários.
Multimedia Metadata Ontology	Fornece um *framework* para a integração de aspectos centrais de metadados multimídia.
Core Ontology for Multimedia	Fornece primitivas para explicitar a composição de um objeto mídia e o que nele deve ser representado. É considerada uma ontologia bem fundamentada para descrição multimídia.
Exif Vocabulary	Especifica formatos a serem usados para imagens e sons em câmaras digitais.
Visual Resources Association	Fornece uma organização categórica para a descrição de trabalhos ligados a cultura visual bem como imagens que os documentam.
Categories for the Description of Works of Art	Descreve objetos de arte e imagens, além de incluir discussões e assuntos relacionados à construção de sistemas de informação no domínio da arte.

Segundo [Schandl et al. 2011], existem muitos vocabulários relevantes para dados multimídia, entretanto, ressaltam que uma grande parte ainda não é utilizada no contexto de *Linked Data*.

5. Considerações finais

Este artigo permitiu evidenciar que há uma quantidade considerável de padrões de metadados, vocabulários e ontologias na tentativa de melhor representar recursos

multimídia visando recuperação semântica através de bibliotecas, portais e bases de dados digitais abertos.

Esforços na construção de ontologias podem ser poupados tendo em vista a exploração de vocabulários em comunidades de interesse. Contudo, surgem desafios na identificação e seleção de uma variedade de padrões de metadados, vocabulários e ontologias disponíveis e que precisam ser compatíveis com as entidades reais de um domínio específico. Tais desafios encontram-se i) no alinhamento de vocabulários e ontologias que reflete aspectos de interoperabilidade semântica e sintática para o provimento de compartilhamento entre sistemas e aplicações na web; e ii) na modelagem conceitual adequada para representar consensualmente parte da realidade de um domínio.

Referências

ALLEMANG, D.; HENDLER, J. (2008) *Semantic web for the working ontologist*: *modeling in RDF, RDFS and OWL*, Elsevier, MA, USA.

BERNERS-LEE, T; HENDLER, J.; LASSILA, O. (2001) "The Semantic Web". *Scientific American*, vol. 284, nº. 5, maio, p. 34-43.

BERNERS-LEE, T. (2006) "Linked Data - Design Issues". Available at: < http://www.w3.org/DesignIssues/LinkedData.html >.

GILLILAND, Anne J. (2000) "Introduction to metadata: setting the stage". Available at:<http://www.getty.edu/research/publications/electronic_publications/intrometadata /setting.pdf >.

GRUBER, T. (1993) "What is an Ontology?" Available at: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.

GUARINO, N. (1998) "Formal ontology in information systems". Available at: <http://citeseer.ist.psu.edu/guarino98formal.html>.

GUIZZARDI, G.; WAGNER, G. (2009) *Using the Unified Foundational Ontology (UFO) as a foundation for general conceptual modeling languages*. Springer-Verlag, Berlin.

JAIN, P.; HITZLER, P.; YEH, P.; VERMA, K.; SHELT, A. (2010) "Linked Data is Merely More Data". *Semantic Technology Conference*. Available at: < http://knoesis.wright.edu/library/publications/linkedai2010_submission_13.pdf >

MASOLO, C.; BORGO, S.; GANGEMI, A.; GUARINO, N.; OLTRAMARI, A. (2003) *Ontology Library*: WonderWeb Deliverable D18. Trento, Italy. Available at: <http://www.loa-cnr.it/Papers/D18.pdf>.

SCHANDL, B.; HASLHOFER, B.; BÜRGER, T.; LANGEGGER, A.; HALB, W. (2011) *Linked Data and multimedia: the state of affairs*. Multimedia Tools and Applications, online first,1-34.

SILVA, D. L. da; SOUZA, R. R.; ALMEIDA, M. B. (2008) "Ontologias e vocabulários controlados: comparação de metodologias para construção". *Ciência da Informação*, v. 37, n.3, p. 60-75.

Registro de procedência de ligações RDF em Dados Ligados

Jonas F. S. M. De La Cerda[1], **Maria Cláudia Cavalcanti**[1]

[1]Instituto Militar de Engenharia
Praça General Tibúrcio, 80 – Praia Vermelha – Rio de Janeiro – RJ

Abstract. *As many tools have been created to support linked data consumption and publishing, there is a demand for quality assessment and to verify these data. To make this possible, data about this consumption should be recorded. This paper presents an extension to a framework with the goal to support the recording and publishing of the information about the creation and consumption of linked data, in order to provide input for later quality assessment.*

Resumo. *Com a criação de ferramentas para consumir, relacionar e publicar dados ligados, surge a demanda para avaliar e comprovar a qualidade destes dados. Para tal, é necessário que informações sobre este consumo sejam registradas. Este trabalho propõe a extensão de uma arquitetura a fim de suportar o registro e publicação de informações sobre a criação destes dados, a fim de prover insumos para posterior avaliação.*

1. Introdução

Com o desenvolvimento e adoção da *web* semântica, vieram padrões e formatos para integrar dados e informações oriundos de diferentes fontes. Há iniciativas para disponibilizar dados em formatos padronizados, para que estes possam ser consumidos (e relacionados) com dados de diferentes fontes. Uma destas iniciativas é o *Linked Data* (dados ligados) [1], que consiste em interligar dados de diversas fontes segundo alguns princípios. Estes princípios são: disponibilizar os dados em um formato padronizado – no caso o RDF (*Resource Description Framework*) [2] – e fornecer meios para acessar e identificar os dados disponibilizados.

É possível criar aplicações mais ricas em informação através do consumo dos dados e seus relacionamentos de diversas fontes. Para tal, é necessário considerar problemas como a obtenção do dado, mapeamento de esquemas e vocabulários, e análise de qualidade do dado. Diante destes problemas, diversas ferramentas foram criadas para facilitar a integração e consumo dos dados ligados, algumas listadas em [Bizer *et al.* 2009]. Não há a preocupação em registrar informações de como estas novas relações foram geradas, criando um problema para provar a confiabilidade e corretude do processo empregado.

Este trabalho propõe uma arquitetura a fim de suportar o registro de informações sobre a criação das interligações de recursos RDF, ou seja, registrar as informações de quais processos foram utilizados para criação, quais parâmetros configuraram estes processos, quais os resultados destes processos. Acredita-se que tais informações podem ajudar em futura análise de qualidade dos dados, tornando-se um ativo tanto para quem con-

[1]http://www.w3.org/DesignIssues/LinkedData.html
[2]http://www.w3.org/TR/REC-rdf-syntax/

some os dados quanto para quem os que publica. A seção 2 deste artigo apresenta os conceitos básicos de dados ligados. A seção 3 apresenta trabalhos relacionados, constando de: uma arquitetura prévia e sua implementação, e modelos de dados de procedência. A seção 4 apresenta a arquitetura proposta, e a seção 5 apresenta as conclusões e extensões do projeto.

2. Dados Ligados

Uma vez que consumir e integrar estes dados se dá de forma mais flexível, é possível escapar do contexto de uma *web* ultrapassada onde aplicações devem prever o consumo de fontes de dados previamente definidos, criando uma *web* onde a informação provida por aplicações pode evoluir ao longo do tempo, junto com o surgimento de novas fontes de dados. Para tirar proveito dos dados ligados, Berners-Lee elucida em um documento [3] regras para publicar (e consumir) os dados ligados: usar URIs válidas para nomear seus recursos (dados, coisas, entidades, etc), de forma que agentes (pessoas ou sistemas) recebam informações úteis – preferencialmente em formato inteligível – ao acessar tais endereços, e, principalmente incluir ligações (links) para recursos em outras fontes de dados, para que novos conhecimentos possam ser descobertos.

Em um tutorial [4] feito por Bizer, define-se uma ligação RDF como uma tripla no formato "sujeito - predicado - objeto" onde o sujeito é ligado ao objeto através de um predicado. As ligações RDF onde o sujeito está em um conjunto de dados e o objeto está em um conjunto de dados distinto são chamados de ligações externas.

3. Trabalhos Relacionados

Existem diversas aplicações utilizando dados ligados. Tais aplicações vão desde *endpoints* SPARQL – formulários onde insere-se uma consulta em SPARQL e recebe-se o resultado da consulta, usualmente no formato de alguma serialização RDF – até aplicações mais complexas como os *websites* da BBC. Em [Kobilarov *et al.* 2009] são apresentados os mecanismos utilizados por estes sistemas a fim de consumir e gerar ligações com outros provedores de dados ligados. São explorados os mecanismos utilizados para interligar os diversos sistemas (legados e atuais) da BBC à nuvem do movimento *Linking Open Data* [5], os mecanismos para reutilização e redirecionamento para conteúdos de outros provedores de dados, os mecanismos da publicação de dados dos programas da emissora.

Em [Bizer *et al.* 2009] é identificada uma arquitetura comum de aplicações voltadas para dados ligados. Tal arquitetura é ilustrada na Figura 1, adaptada de [Isele *et al.* 2010], excluindo-se a parte tracejada da figura, que representa um coletor de dados de procedência a ser explicado mais adiante. Para consumir – importar, associar e publicar – os dados ligados da *web*, uma aplicação tem que considerar problemas como obtenção do dado, mapeamento de esquemas e vocabulários e análise de qualidade do dado. Existe uma implementação funcional de um arcabouço para executar todas as etapas da integração dos dados ligados previstas pela arquitetura comum, o LDIF (*Linked Data Integration Framework*) [Schultz *et al.* 2011].

[3] http://www.w3.org/DesignIssues/LinkedData.html

[4] http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

[5] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

Ao passo que o LDIF ataca os problemas de mapeamento de esquemas e vocabulários, resolução de identidades, importação, publicação e descoberta de ligações (relações entre recursos), o arcabouço se apresenta deficiente no quesito da procedência dos dados. Procedência refere-se à linhagem dos dados, isto é, as origens e histórico de processamento de objetos e processos [Bose e Frew 2005], ou seja, a procedência possui um papel importante em evidenciar a qualidade dos dados gerados.

A deficiência do LDIF quanto à captura da procedência é evidente pois os únicos dados de procedência publicados são os dados relativos à importação inicial dos dados, ou seja, qual a origem dos dados importados. Dados importantes de procedência como a parametrização de processos de similaridade sintática e semântica, resultados da execução de processos, dentre outros, não são contemplados, nem pelo LDIF e nem pela arquitetura de aplicações de dados ligados.

Os dados de procedência podem servir de insumo para análise de qualidade dos dados gerados. Pode-se atribuir maior confiabilidade a dados gerados por processos que foram configurados com limites mais restritos. Por exemplo, é possível atribuir maior confiabilidade às ligações geradas por processos de cálculo de similaridade que tenham sido configurados com um limite de similaridade maior que 0.95 (95%). Em [Mendes *et al.* 2012] são ilustrados tanto exemplos de avaliação de qualidade dos dados quanto de fusão de dados. Um dos exemplos mostrados por Mendes, é a atribuição de reputação aos dados de acordo com sua origem, e, a pontuação (*scoring*) de acordo com o quão recente o dado é.

Dada a importância dos dados de procedência, alguns modelos influenciaram este trabalho. O mais notável é o OPM (*Open Provenance Model*) [Moreau *et al.* 2011], que descreve as relações causais e de dependência entre artefato (que representa o estado imutável de um objeto), processo (que representa ações efetuadas em um artefato, ou causadas por) e agente (que representa entidades que podem facilitar, controlar ou influenciar um processo de alguma forma). Os outros modelos que influenciaram este são o *Provenir* [Sahoo e Sheth 2009] e o PROV-DM [6]. Os conceitos definidos pelo OPM estão presentes também nestes modelos. No caso do *Provenir*, estes conceitos são mais especializados (e.g. diferenciação de dados e parâmetros). Já o PROV-DM não é tão específico quanto aos artefatos, porém possui muitas definições das relações de dependência e causalidade, inclusive sendo especificadas formalmente.

4. Arquitetura Proposta

Este trabalho propõe que a arquitetura das aplicações ainda deficiente na questão da procedência de dados contemple tal aspecto, fornecendo um modelo de dados para o processo de integração de dados ligados. A arquitetura deve contemplar o aspecto de procedência em todas as etapas dos processos de consumo e integração, conforme mostra a Figura 1. Para tal, diversos modelos de procedência devem ser estudados, a fim de definir um modelo que seja compatível com os modelos já existentes e difundidos.

O modelo de procedência a ser adotado na nova arquitetura deve não somente contemplar a diferenciação entre dados e parâmetros, mas também deve diferenciar os processos empregados na integração dos dados ligados, considerando a hierarquia

[6] http://www.w3.org/TR/prov-dm/

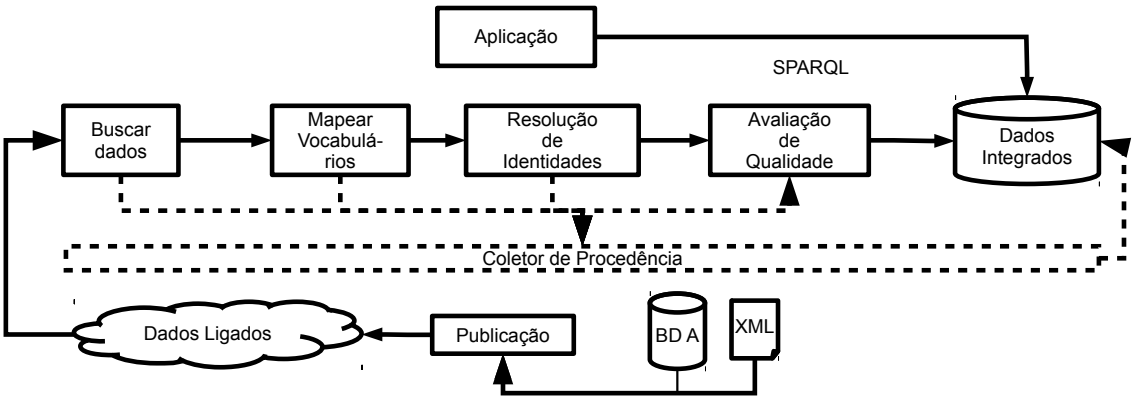


Figura 1. Arquitetura de aplicações consumidoras de dados ligados considerando os aspectos de procedência de dados.

de técnicas empregadas tanto no mapeamento de vocabulários quanto na descoberta de links. Uma visão de como essas técnicas podem ser classificadas foi apresentada por [Euzenat e Shvaiko 2007] e foram também estudadas por [Silva 2010], que relacionou esta visão com as medidas de similaridades definidas por [Ehrig 2007].

Até o momento, o modelo considera alguns aspectos básicos quanto aos tipos de processos utilizados na integração e consumo de dados ligados, e, considera uma categorização dos dados em questão. Os tipos de processo contemplados até o momento são processos de importação – processos que obtém os dados de seus provedores originais – e processos de integração. Os processos de integração se encontram categorizados como processos de mapeamento (de vocabulários) e processos de ligação.

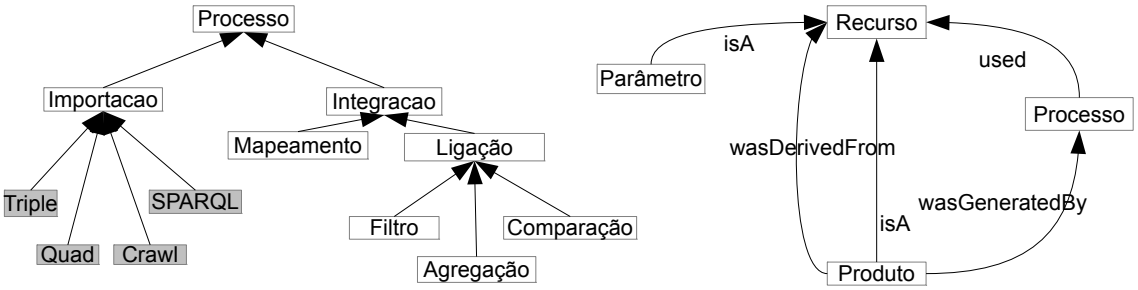


Figura 2. Conceitos do modelo de dados de procedência.

Os processos de mapeamento tratam-se de definições de pareamentos de conceitos de um vocabulário a outro, como parear foaf:Person e dbpedia:Person ou foaf:name e rdfs:label. Os processos de ligação tratam-se de execuções de processos que geram ligações RDF através de alguma computação. Tais processos podem ser processos de agregação – como médias, máximos, mínimos – processos de filtragem – como selecionar recursos que possuam uma determinada propriedade dentro de um intervalo de valores – e processos de comparação – como comparar rótulos RDF através de distância de edição, ou comparar a categorização de dois recursos.

Pode-se dizer que há uma equivalência entre os conceitos de processo do OPM e processo do modelo proposto. Uma ideia inicial do modelo é ilustrada pela Figura 2,

onde os conceitos em cinza-escuro representam extensões dos processos de importação, inclusive já implementados no LDIF.

No que concerne ao conceito de artefato do OPM, há uma relação de equivalência com o conceito de recurso, subcategorizado em parâmetro e produto, como mostra a Figura 2. A diferença entre produtos e parâmetros é que produtos são gerados por processos, ou seja, para gerar cada produto foram consumidos tempo e recursos computacionais.

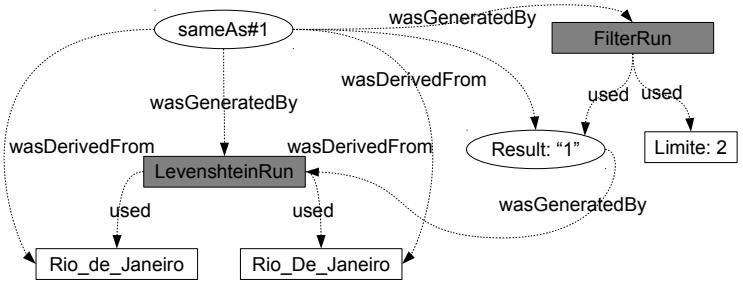


Figura 3. Exemplo de aplicação de modelo.

A Figura 3 exemplifica uma aplicação bastante básica do modelo, a criação de uma ligação do tipo "owl:sameAs" entre dois recursos de rótulos "Rio_de_Janeiro" e "Rio_De_Janeiro", respectivamente. A geração da ligação se dá em dois passos, o primeiro sendo a comparação entre os rótulos dos recursos através de um algoritmo que calcula distância de edição entre duas cadeias de caracteres e o segundo filtrando apenas os produtos que tenham sido gerados com distância de edição abaixo de 2. Na Figura 3, os produtos estão representados por elipses, os parâmetros por retângulos claros e os processos por retângulos escuros. Explicitar todas as relações causais entre dados e processos pode gerar um excesso de informações, que é problema conhecido e já foi discutido em [Heinis e Alonso 2008], não sendo o foco deste trabalho.

Em resumo, modelo e arquitetura propostos encapsulam os executores dos processos envolvidos em cada etapa do fluxo da integração e consumo de dados ligados, a fim de registrar e representar os dados de procedência de acordo com a natureza dos processos envolvidos na criação das ligações RDF entre recursos, bem como a natureza dos parâmetros que configuram estes processos e resultados destes processos. Dessa forma, esses dados de procedência passam a estar disponíveis para um usuário avaliar confiabilidade e autenticidade das ligações geradas, avaliar a qualidade e efetuar fusão de dados ligados – como é o caso do Sieve [Mendes *et al.* 2012] – e reproduzir o processo de geração de ligações RDF.

5. Conclusão

Este artigo apresenta uma proposta para o problema do registro e representação de procedência de dados na atividade de integração e consumo de dados ligados. A sua principal contribuição é a extensão de modelos de procedência já estabelecidos e ainda em definição, adaptando-os para registrar informações mais específicas sobre o consumo e integração de dados ligados. A partir de uma arquitetura já existente – o LDIF – de código aberto, estende-se sua funcionalidade de modo a suportar o registro dessas informações. No momento a extensão proposta está em fase de implementação. O modelo de dados

proposto ainda passa por refinamentos, devendo evoluir a fim de especificar os processos envolvidos e tipos de dados e parâmetros.

Trabalhos futuros incluem o estabelecimento de políticas de descarte e seleção de ligações RDF, com base nos dados de procedência disponibilizados. Além disso, conforme as ligações RDF são rastreadas e associadas às informações de procedência, é possível estabelecer e configurar mecanismos de inferência baseados nessas informações.

Acknowledgements

The authors would like to thank CNPq (309307/2009-0; 486157/2011-3) and FAPERJ (E-26/111.147/2011) for partially funding their research projects.

Referências

Bizer, C., Heath, T., e Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.

Bose, R. e Frew, J. (2005). Lineage retrieval for scientific data processing: a survey. *ACM Computing Surveys*, 37:1–28.

Ehrig, M. (2007). *Ontology Alignment: Bridging the Semantic Gap*, volume 4 of *Semantic Web And Beyond Computing for Human Experience*. Springer.

Euzenat, J. e Shvaiko, P. (2007). *Ontology matching*. Springer.

Heinis, T. e Alonso, G. (2008). Efficient lineage tracking for scientific workflows. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1007–1018, New York, NY, USA. ACM.

Isele, R., Jentzsch, A., e Bizer, C. (2010). Silk server - adding missing links while consuming linked data. In *1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai*.

Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., e Lee, R. (2009). Media meets semantic web — how the bbc uses dbpedia and linked data to make connections. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 723–737, Berlin, Heidelberg. Springer-Verlag.

Mendes, P. N., Mühleisen, H., e Bizer, C. (2012). Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, pages 116–123, New York, NY, USA. ACM.

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P. T., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E. G., e den Bussche, J. V. (2011). The open provenance model core specification (v1.1). *Future Generation Comp. Syst.*, 27(6):743–756.

Sahoo, S. S. e Sheth, A. (2009). Provenir ontology: Towards a framework for escience provenance management. Microsoft eScience Workshop.

Schultz, A., Matteini, A., Isele, R., Bizer, C., e Becker, C. (2011). *LDIF - Linked Data Integration Framework*, pages 1–4.

Silva, V. d. S. (2010). Uma abordagem para alinhamento de ontologias biomédicas para apoiar a anotação genômica. Master's thesis, Universidade Federal do Rio de Janeiro.

Descoberta Automática de Relações Não-Taxonômicas a partir de Corpus em Língua Portuguesa

Vinicius H. Ferreira, Lucelene Lopes, Renata Vieira

PPGCC – FACIN – Porto Alegre – Brasil

vinihf@gmail.com, {lucelene.lopes,renata.vieira}@pucrs.br

Resumo. *A construção de ontologias é um processo complexo que compreende etapas como a extração de conceitos de domínio e a extração de relações taxonômicas e não-taxonômicas entre esses conceitos. A etapa de extração de relações não-taxonômicas é a mais negligenciada, especialmente para textos em língua portuguesa. Com isto, este trabalho apresenta uma proposta de extração de relações não-taxonômicas a partir de textos em português representados por uma lista de conceitos e informações contextuais automaticamente extraídos pela ferramenta ExATOlp.*

Abstract. *Ontology construction is a complex process composed by extraction tasks for domain concepts, as well as taxonomic and non-taxonomic relations among concepts. The extraction of non-taxonomic relations is the most neglected task, specially for Portuguese texts. Therefore, this paper presents a proposal for extracting non-taxonomic relations from Portuguese texts represented by a list of concepts and their contextual information extracted automatically by ExATOlp software tool.*

1. Introdução

A construção de ontologias de domínio [Gruber, 1993] é um ramo relevante da área de Processamento de Linguagem Natural (PLN). Tradicionalmente todo o processo de construção de ontologia é dependente de especialistas do domínio estudado. No entanto, esses especialistas são na maioria das vezes sobrecarregados pelo tamanho e complexidade dos dados e informações contidos no processo de construção de ontologias [Cimiano et al. 2006]. Com isto, busca-se métodos automáticos que reduzam a demanda de intervenção de especialistas [Maedche e Staab, 2000]. Dentre esses métodos, o presente trabalho interessa-se por aqueles que constroem ontologias a partir de *corpus*, que por sua vez é um conjunto de textos sobre um domínio específico [Biemann, 2005]. Dentre vários trabalhos [Maedche e Staab, 2000; Sanchéz e Moreno, 2008; Serra e Girardi, 2011; Villaverde et al. 2009; Schutz e Buitelaar, 2005], o processo proposto por Lopes (2012) gera, como uma das saídas possíveis, uma lista de conceitos e as informações contextuais da utilização de cada conceito no domínio. Dentre estas informações são identificadas a função sintática e os verbos aos quais o conceito se relaciona.

Para Maedche e Staab (2000), o processo de construção de ontologias contempla três etapas básicas: (i) extração de conceitos de domínio; (ii) extração de taxonomia; e (iii) extração de relações não-taxonômicas. A maior parte dos trabalhos

semelhantes coleta conceitos relevantes de um domínio e pode agrupá-los em uma hierarquia (taxonomia) utilizando métodos linguísticos e estatísticos [Lopes, 2012; Pantel e Lin, 2001; Chung, 2003; Brewster et al. 2003]. De acordo com Sanchéz e Moreno (2008), no processo de Aprendizagem de Ontologias, a fase de extração de relações não-taxonômicas tem sido reconhecida como a mais complexa e negligenciada [Maedche e Staab, 2000; Sanchéz e Moreno, 2008; Villaverde et al. 2009].

Diferente das relações taxonômicas, que contribuem na estruturação de um domínio e classificação de conceitos, as relações não-taxonômicas não estão relacionadas a hierarquia. Este tipo de relação acrescenta informações aos conceitos já encontrados, identificando relacionamentos entre eles [Guarino e Welty, 2002].

Identificar as relações não-taxonômicas é essencial para expressar as propriedades de classes e entidades de um domínio específico [Cimiano et al. 2006], representando as ações ou eventos que ocorrem entre os conceitos. Por exemplo, uma relação não-taxonômicas no campo do Direito, é a relação "representa" entre os conceitos "Advogado" e "Cliente" [Serra e Girardi, 2011], e no campo do Esporte, a relação "chuta" entre os conceitos "Jogador" e "Bola" [Schutz e Buitelaar, 2005].

De acordo com Serra e Girardi (2011), relações não-taxonômicas podem ser classificadas como independentes e dependentes de domínio. As relações independentes de domínio podem ser divididas em: (i) agregação, identificadas por relações "todo-parte"; e (ii) propriedade, identificadas por relações de posse ou composição. Relações dependentes de domínio são identificadas por termos específicos de um domínio.

O papel dos verbos como elemento de conexão central entre conceitos é inegável. Eles são responsáveis por especificar qual é a interação entre os participantes de uma ação ou evento, expressando a relação entre eles. Devido a isto os verbos tem sido muito utilizados para definir relações não-taxonômicas [Kavalec et al. 2004; Maedche e Staab, 2000; Sanchéz e Moreno, 2008; Schutz e Buitelaar, 2005].

Partindo disso, esse trabalho apresenta uma proposta de processo para extração de relações não-taxonômicas em textos na língua portuguesa. Diferente dos trabalhos similares de Sanchéz e Moreno (2008) e Brewster et al. (2003), aqui utiliza-se como fonte a lista de conceitos e informações contextuais geradas pelo ExATOlp que é uma ferramenta que implementa todas etapas do processo de extração de conceitos e contextos proposto por Lopes (2012).

2. Trabalhos Similares

Na literatura encontram-se trabalhos que propõe processos de extração de relações não-taxonômicas a partir de textos. A Tabela 1 apresenta uma síntese a respeito dos trabalhos similares ao apresentado neste artigo.

Pode-se observar através da Tabela 1, com exceção do trabalho de Sanchéz e Moreno (2008), todos os demais utilizam um conjunto de textos não-estruturados de um determinado domínio (*corpus* de domínio) como fonte para o processo de extração de relações não-taxonômicas. Dentre esses trabalhos, com exceção do trabalho de Maedche e Staab (2000), todos utilizam o verbo como elemento principal na identificação de relações não-taxonômicas. Dentro deste contexto, observou-se a constante ocorrência da manipulação da tripla *<conceito 1, conceito 2, verbo>* nos trabalhos apresentados.

Embora nem todos os trabalhos tenham seu foco em encontrar relações não-taxonômicas em corpus da língua inglesa, nenhum deles apresenta uma proposta para a língua portuguesa. Além disso, foi possível observar que na maioria dos trabalhos analisados as relações não-taxonômicas extraídas são dependentes do domínio, ou seja, a relação entre os conceitos é feita por termos específicos do domínio. Uma possível justificativa para isso é que nestes trabalhos os verbos que relacionam os conceitos são também usados como identificadores da relação.

Tabela 1. Comparação de trabalhos similares

Trabalhos Similares	Fonte de dados	Relação identificada por verbo?	Idioma	Tipos de relações não-taxonômicas extraídas
Maedech e Staab (2000)	*Corpus* de domínio	Não	Alemão	Dependente de domínio
Schutz e Buitelaar (2005)	*Corpus* de domínio	Sim	Inglês e Alemão	Dependente de domínio
Sanchéz e Moreno (2008)	Web	Sim	Inglês	Dependente de domínio
Villaverde et al. (2009)	*Corpus* de domínio, lista de candidatos a conceitos ou hierarquia de conceitos	Sim	Inglês	Dependente de domínio
Weichselbraun et al. (2009)	Ontologia e *corpus* de domínio	Sim	Inglês	Independente de domínio
Serra e Girardi (2011)	*Corpus* de domínio	Sim	Inglês	Dependente e independente de domínio

Na análise dos trabalhos similares foi possível observar que a maioria das propostas não se propõe a identificar relações não-taxonômicas de forma automática, mas sim sugerir relações para especialistas de domínio. Com isso, pode-se verificar que o papel dos engenheiros de ontologias e especialistas de domínio é importante na decisão final a respeito de relações não-taxonômicas [Serra e Girardi, 2011].

3. Processo Proposto

O processo de extração de relações não-taxonômicas apresentado neste artigo tem seu foco em conceitos da língua portuguesa, e ele constitui-se de 5 etapas distintas (Figura 1): (i) Aquisição dos termos com informações contextuais; (ii) Eliminação de termos com informações faltantes; (iii) Identificação dos conceitos do domínio; (iv) Extração dos candidatos a relações não-taxonômicas; e (v) Visualização das relações extraídas.

Conforme verificou-se através dos trabalhos similares, os verbos são os elementos fundamentais na identificação de relações não-taxonômicas entre conceitos de um domínio. É válido também salientar que na língua portuguesa uma oração tem como estrutura sintática básica "Sujeito + Verbo + Objeto". Constatou-se então que a tabela de termos com informações contextuais produzida pela ferramenta ExATOlp provê todas informações necessárias para a extração de relações não-taxonômicas. Sendo assim, na primeira etapa do processo é feita a aquisição dos termos e todas as suas informações contextuais da lista produzida pelo ExATOlp.

Embora na etapa de aquisição sejam extraídas todas informações contextuais dos termos, para a extração de relações não-taxonômicas são utilizados apenas o termo, sua função sintática (sujeito ou objeto) e o verbo na forma canônica com o qual ele se relaciona. A aquisição de todas informações contextuais é feita com o objetivo de prover informações adicionais aos conceitos que se relacionam para o desenvolvimento de futuras aplicações linguísticas.

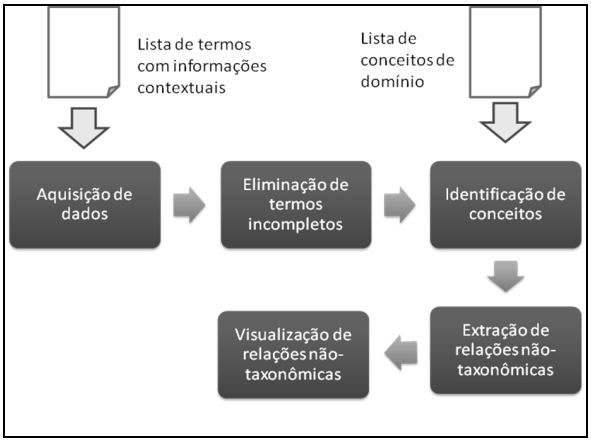


Figura 1. Processo de extração de relações não-taxonômicas

Na segunda etapa são eliminados todos os termos que não possuem verbos associados, ou que o processo de extração não conseguiu identificar a função gramatical. Esta é uma etapa de importante para a rapidez de execução do processo, pois evita que trabalhe-se com termos incompletos. Como o objetivo do processo é extrair relações não-taxonômicas entre conceitos, é necessário processar apenas os termos que foram considerados conceitos de um domínio. Para isto, na terceira etapa do processo os termos são comparados aos conceitos extraídos pelo processo proposto por Lopes et al. (2012) e elimina-se todos os termos que não são classificados como conceitos.

Na etapa de extração de relações não-taxonômicas, são identificados os conceitos que se relacionam através de um mesmo verbo. Estes conceitos são então classificados pela sua função gramatical como sujeitos ou objetos. São definidas então triplas no formato < *Conceito Sujeito, Verbo, Conceito Objeto* > produzidas pelo produto cartesiano entre os conceitos considerados sujeitos e objetos e o verbo que os relaciona. As triplas definidas nesta etapa representam candidatos a relações não-taxonômicas do domínio. Para que seja possível explorar as relações extraídas pelo processo, a última etapa tem seu foco em permitir que usuários (especialistas) visualizem-nas de forma simplificada.

4. Experimento

Com o objetivo de avaliar o funcionamento do processo proposto, foi desenvolvido um sistema computacional para operacionalizar cada uma de suas etapas. Através do sistema desenvolvido, foi realizado um experimento com uma lista de termos, conceitos e informações contextuais produzida pelo ExATOlp a partir de um *corpus* de Geologia. Este *corpus* contém 234 textos, 69.461 frases e 2.010.527 palavras.

Na execução da etapa de aquisição de dados foram encontradas 255.816 ocorrências de termos com informações contextuais. Com o objetivo de eliminar os

termos com informações faltantes, foi executada a segunda etapa do processo, restando 68.831 ocorrências. Na terceira etapa, os termos foram comparados com os conceitos do domínio identificados pelo ExATOlp. Todos os termos que não constavam nessa lista de conceitos foram eliminados, restando então 18.025 ocorrências de conceitos com suas informações contextuais. Sobre estas ocorrências foi realizada a quarta etapa do processo, que produziu 270.197 triplas candidatas a instâncias de relações não-taxonômicas do domínio de Geologia que correspondem a 418 relações distintas.

A quinta e última etapa não foi experimentada, porém para que fosse possível visualizar as triplas candidatas foi desenvolvida uma aplicação Web a ser utilizada pelos especialistas do domínio. Essa aplicação apresenta os dados na forma de um dicionário, permitindo a exploração das instâncias das relações. A Figura 2 apresenta um exemplo onde visualiza-se o conceito "evento vulcânico" (sujeito) e as instâncias (cada objeto) da relação "provocar" (verbo), ou seja, tudo que "eventos vulcânicos" pode "provocar".

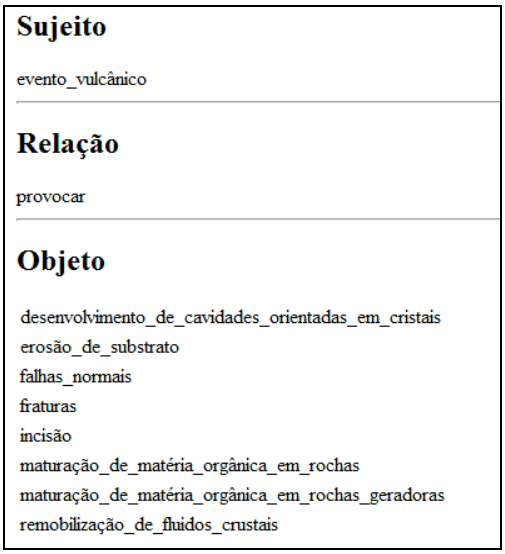


Figura 2. Visualização das relações não-taxonômicas extraídas

8. Conclusões

Conforme pode ser visto através da execução do experimento com o *corpus* de Geologia, o processo proposto permite a extração de relações não-taxonômicas tendo por base a lista de conceitos e suas informações contextuais gerada pelo ExATOlp, ou seja, as informações contextuais dos conceitos disponibilizadas pelo ExATOlp provêem dados suficientes para a descoberta de relações não-taxonômicas.

Um trabalho futuro será avaliar a relevância das relações extraídas para o domínio. Dessa forma, assim como em trabalhos similares, é necessário uma etapa de avaliação por especialistas do domínio das relações extraídas. Portanto, as 418 relações extraídas no processo proposto serão consideradas candidatas e somente após esta etapa de avaliação serão consideradas relações não-taxonômicas do domínio. Outra possibilidade é o uso de pontos de corte, a exemplo do que já foi desenvolvido para selecionar conceitos [Lopes et al. 2010].

Outro trabalho futuro planejado é aplicação do processo proposto para outros *corpora*, como os utilizados por Lopes (2012) relativos às áreas de Pediatria, Modelagem Estocástica, Mineração de Dados e Processamento Paralelo.

Referências

Biemann, C. (2005) *Ontology learning from text: a survey of methods*. LDV Forum, 20: 75-96.

Brewster, C.; Ciravegna, F.; Wilks, Y. (2003) *Background and foreground knowledge in dynamic ontology construction*. Proc. of the SIGIR Semantic Web Workshop.

Chung, T. M. (2003) *A corpus comparison approach for terminology extraction*. Terminology, 9: 221-246.

Cimiano, P.; Volker, J.; Studer, R. (2006) *Ontologies on demand? - a description of the state-of-the-art, applications, challenges and trends for ontology learning from text*. Information, Wissenschaft und Praxis, 57: 315-320.

Gruber T. (1993) *Toward principles for the design of ontologies used for knowledge sharing*. International Journal Human-Computer Studies, 43: 907-928.

Guarino, N. e Welty, C. (2002) *Evaluating ontological decisions with OntoClean*. Communications of the ACM, 45(2): 61-65.

Kavalec M.; Maedche, A.; Svátek, V. (2004) *Discovery of lexical entries for non-taxonomic relations in ontology learning*. Proc. of Int. Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM), LNCS 2932: 249-256.

Lopes, L.; Vieira, R.; Finatto, M. J.; Martins, D. (2010) *Extracting compound terms from domain corpora*. Journal of the Brazilian Computer Society, 16(4): 247-259.

Lopes, L. (2012) Extração automática de conceitos a partir de textos na língua portuguesa. 156 f. Tese de doutorado em Ciência da Computação - FACIN, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.

Lopes, L.; Fernandes, P.; Vieira, R. (2012) Domain term relevance through tf-dcf. Proc. of Int. Conf. on Artificial Intelligence (ICAI).

Maedche A. e Staab S. (2000) *Mining non-taxonomic conceptual relations from text*. Proc. of the 12th European Knowledge Acquisition Workshop, Juan-les-Pins.

Pantel, P. e Lin, D. (2001) *A statistical corpus-based term extractor*. Proc. of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, ACM Press: 36-46.

Sanchéz, D. e Moreno, A. (2008) *Learning non-taxonomic relationships from web documents for domain ontology construction*. Data & Knowledge Eng, 64: 600-623.

Serra, I. e Girardi, R. (2011) *A process for extracting non-taxonomic relationshop of ontologies from text*. Intelligent Information Management, 3: 119-124.

Schutz, A. e Buitelaar, P. (2005) RelExt: *A tool for relation extraction in ontology extension*. Proc. of the Fourth Int. Semantic Web Conference: 593-606.

Villaverde, J.; Persson, A.; Godoy, D.; Amandi, A. (2009) *Supporting the discovery and labeling of non-taxonomic relationships in ontology learning*. Experts Systems with Applications, 36: 10288-10294.

Weichselbraun, A.; Wohlgenannt, G.; Scharl, A.; Granitzer, M.; Neidhart, T.; Juffinger, A. (2009) *Discovery and evaluation of non-taxonomic relations in domain ontologies*. Int. Journal of Metadata, Semantics and Ontologies, 4(3): 212-222.

Uma Proposta para o Uso de Folksonomias como Conceitualizações Compartilhadas na Especificação de Modelos Conceituais

Josiane M. P. Ferreira[1,2], Cesar Augusto Tacla[1], Sérgio R. P. da Silva[2]

[1]CPGEI – Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de setembro 3165, CEP 80230-901, Curitiba-PR

[2]Departamento de Informática – Universidade Estadual de Maringá
Av. Colombo 5.790, CEP 87020-900, Maringá-PR

josiane@gmail.com, tacla@utfpr.edu.br, sergio.r.dasilva@gmail.com

Abstract. *This work purposes to use data of collaborative tagging as shared conceptualization that can be useful to build conceptual models. The hypothesis assumes that the folksonomy induced from the collaborative tagging's data based on parameters of authorship and motivation of categorization can represent a shared conceptualization of a domain. Thus, it is expected that the utilization of this folksonomies generate the decrease of the divergences in the terms elicitation that will be part of the conceptual model when compared with algorithms of folksonomy induction that don't use this parameters.*

Resumo. *Este trabalho propõe utilizar os dados de tagging colaborativo como conceitualizações compartilhadas que possam ser úteis na construção de modelos conceituais. A hipótese adotada é a de que a folksonomia induzida dos dados de tagging colaborativo com base nos parâmetros de autoria das tags e motivação das categorizações represente uma conceitualização compartilhada de domínio. Desta forma, espera-se que a utilização destas folksonomias provoque a diminuição de divergências na elicitação de termos que farão parte do modelo conceitual em comparação com algoritmos de indução de folksonomias que não utilizam estes parâmetros.*

1. Introdução

Guizzardi (2005, p. 2) adota o nome **conceitualização** para designar o conjunto de conceitos utilizados para articular abstrações do estado das coisas em um domínio. **Modelo** é uma abstração de uma porção da realidade articulada segundo uma conceitualização de um domínio. Ainda, para Guizzardi (2005), tanto conceitualizações como modelos existem somente nas mentes das pessoas. O que há de concreto são **especificações de modelos conceituais** feitas em uma **linguagem de modelagem** que permitem expressar (representar) conceitualizações. Desta forma, a especificação do modelo conceitual – denominada de modelo conceitual, é um artefato concreto que permite aos atores envolvidos no processo de construção do modelo compreender o domínio, atingir consenso sobre o significado das entidades representadas e se comunicar. Deste modo, uma **ontologia**, do ponto de vista de artefato, é um caso particular de modelo concreto.

Na passagem das conceitualizações e modelos abstratos para modelos concretos ocorre o problema descrito por Feigeinbaum (1984) denominado de **gargalo de aquisição de conhecimentos** que diz respeito à dificuldade que os engenheiros de conhecimentos têm em capturar e representar conhecimentos a partir de interações com especialistas. As ontologias, modelos concretos destinados a comunidades de usuários, foco particular deste artigo, necessitam de uma aquisição de conhecimentos que envolvem também um grande número de atores, além de outras fontes de informação.

Realizar aquisição de conhecimentos em larga escala é demorado e custoso. Atingir consenso com um número elevado de atores torna-se difícil, pois aumentam as divergências, assim como o número de interações para resolvê-las. Há abordagens de aprendizado de ontologias que se utilizam de métodos e técnicas de processamento de linguagem natural, aprendizado de máquina e mineração de textos para extrair conceitos, relações e instâncias de fontes de informação processáveis (ex. *schemas* de bancos de dados, textos) [Maedche e Staab 2001]. Algumas destas abordagens têm utilizado dados dos sistemas baseados em *tagging* colaborativo como fonte de informação para estes algoritmos.

Sistemas de *tagging* colaborativo são aplicações ditas sociais que permitem aos seus usuários atribuírem etiquetas (*tag*s) a recursos da Web. Um recurso pode ser etiquetado por vários usuários com as *tags* que acharem convenientes. O fato interessante é que, apesar de não existir um vocabulário controlado, depois de certo tempo as *tags* utilizadas pelos usuários para etiquetar um recurso parecem se estabilizar [Robu, Halpin e Shepherd 2009]. Este trabalho pretende utilizar os dados do *tagging* colaborativo para induzir estruturas, denominadas de folksonomias, que possam ser consideradas como representativas de conceitualizações compartilhadas de um domínio. A hipótese adotada é a de que estruturas que emergem da dimensão social do *tagging* atenuam o gargalo de aquisição que ocorre na especificação de modelos conceituais de domínios, por representarem uma conceitualização compartilhada em uma comunidade de usuários.

Sendo assim, o objetivo deste trabalho é determinar se as folksonomias que emergem dos dados do *tagging* colaborativo são úteis na construção de modelos conceituais. Especificamente, pretende-se construir um algoritmo que leve em conta informações de autoria das *tags* e de motivação de etiquetagem, e avaliar se as folksonomias produzidas com base nestes parâmetros realmente auxiliam a atenuar o gargalo da aquisição de conhecimento na construção de modelos conceituais. Espera-se que a utilização destas folksonomias provoque a diminuição de divergências na elicitação de termos que farão parte de um modelo conceitual em comparação com algoritmos de indução de folksonomias que não utilizam estes parâmetros.

A seção 2 justifica o uso dos dados de *tagging* colaborativo e comenta sobre outras abordagens neste sentido. A seção 3 discute como este trabalho pretende induzir folksonomias como conceitualizações compartilhadas dos dados de *tagging* colaborativo. A seção 4 descreve a metodologia e a seção 6 comenta as contribuições desta proposta.

2. Abordagens que utilizam dados de *tagging* colaborativo

Em sistemas baseados em *tagging* os usuários podem associar quantas e quais *tags* quiserem para um recurso. Ao associarem as mesmas *tags* aos mesmos recursos, os

usuários constroem um "vocabulário consensual" para um determinado conjunto de recursos que pode ser representativo da conceitualização de um domínio. Este fato é mencionado por vários autores, tais como, Robu, Halpin e Shepherd (2009), Angeletou *et al.* (2007), Jäschke *et al.* (2008), Mika (2007), entre outros.

Alguns autores chamam os dados do *tagging* colaborativo de folksonomia. Neste artigo, o termo folksonomia designa a **estrutura** coletiva (lista de termos, taxonomia, categorização) que emerge do *tagging* colaborativo por meio de um algoritmo de indução de folksonomias [Strohmaier *et al.* 2012].

Os termos resultantes do *tagging* colaborativo carregam uma dimensão social de uso. Por isso, várias abordagens de aprendizado de ontologias, ou que simplesmente objetivam derivar alguma taxonomia ou conjunto de conceitos, utilizam estes dados para construir algum tipo de estrutura "consensual" a partir destes dados. As *tags* derivadas do *tagging* colaborativo apresentam uma estrutura plana, ou seja, a única relação explícita entre duas *tags* é a relação de coocorrência – duas *tags* coocorrem se elas fazem parte de uma mesma etiquetagem. Como esta é a única relação entre duas *tags* várias abordagens que identificam alguma estrutura coletiva das *tags* a utilizam como ponto de partida. Entre estas abordagens encontram-se: Begelman, Keller e Smadja (2006), X. Wu, Zhang e Yu (2006), Jäschke *et al.* (2008), Schmitz (2006), Mika (2007), Cattuto *et al.* (2008), Specia e Motta (2007), Angeletou *et al.* (2007) e Hamasaki *et al.* (2007).

O fato é que várias destas abordagens **pressupõem** que as folksonomias ajudam no desenvolvimento de modelos consensuais pelo fato de resultarem de um processo humano e coletivo sem, no entanto, verificar com profundidade a natureza do conhecimento existente no *tagging* (quem o fez, ou por qual motivo, por exemplo). A maioria das abordagens citadas procura avaliar o algoritmo utilizado que induz a estrutura coletiva dos dados de *tagging*, sem, no entanto, avaliar a utilidade da estrutura derivada, ou a origem dos dados de entrada.

3. Folksonomias como conceitualizações compartilhadas

Praticamente nenhuma das abordagens de indução de folksonomias citadas avalia a origem dos dados do *tagging*, como, por exemplo, qual o conhecimento/especialidade do usuário que fez a etiquetagem e o motivo que o levou a etiquetar.

A **motivação** do usuário ao realizar uma etiquetagem pode ser reveladora do significado pretendido para a *tag*, o que é importante no momento de se construir um modelo conceitual. Neste trabalho, defende-se a ideia de que a motivação para criar uma *tag* tem influência no seu uso (ou não) durante a criação de um modelo conceitual. Körner *et al.* (2010) abordam a motivação dos usuários durante a etiquetagem e tentam identificá-la automaticamente separando-as em dois grandes grupos: *tags* de categorização e *tags* de descrição de recursos. Quando as *tags* são utilizadas para categorizar, há pouco uso de sinônimos (o que deve facilitar o consenso entre os atores envolvidos na especificação do modelo conceitual) e a estrutura induzida dos dados de *tagging* se aproxima de uma taxonomia. Quando as *tags* são utilizadas para descrever recursos, então há uso mais proeminente de sinônimos e o vocabulário é, portanto, frequentemente maior, dificultando o consenso na especificação do modelo conceitual.

Outro ponto importante a ser considerado é saber quem realizou a etiquetagem. Segundo Wilson (1983), entidades consideradas autoridades em determinado assunto

tendem a organizar melhor suas informações, possuírem conteúdos de qualidade e manterem contato com pessoas que entendam ou tenham interesse no mesmo assunto. O autor define o conceito de **autoridade cognitiva –** uma autoridade fundamentada na competência e nas capacidades intelectuais de quem a recebe e cuja concessão é compreendida como o reconhecimento e o mérito por estas capacidades – uma autoridade que define "quem sabe o quê sobre o quê". Desta forma, acredita-se que os dados de *tagging* elaborados por usuários que são considerados autoridade cognitiva sobre o domínio de interesse tendem a ser mais informativos sobre o domínio do que os dados de *tagging* elaborados por um usuário leigo.

As abordagens de X. Wu, Zhang e Yu (2006), Jäschke *et al.* (2008), Schmitz (2006), Mika (2007) e Hamasaki *et al.* (2007) para derivar estruturas dos dados de *tagging* colaborativo utilizam informações sobre a autoria das *tags* (em termos de qual usuário utilizou qual *tag* para etiquetar qual recurso) para extrair a relação de coocorrência entre as *tags*, mas sem avaliar o conhecimento do usuário sobre o recurso que está sendo categorizado.

Portanto, propõe-se levar em consideração a autoria das *tags* (em termos de autoridade cognitiva) e a motivação na criação das mesmas para melhor utilizar as folksonomias como fonte de informação na construção de modelos conceituais.

4. Metodologia proposta

Um algoritmo para induzir folksonomias que considere a autoria e a motivação das etiquetagens a serem utilizadas na modelagem conceitual está sendo construído. O algoritmo deve selecionar *tags* sobre o domínio para o qual se pretende construir o modelo conceitual. Estas *tags* devem ter sido criadas por usuários categorizadores e considerados autoridades no domínio em questão. Para fins de comparação, um algoritmo de indução de folksonomias que não utiliza as informações de autoria e de motivação, servirá de referência na avaliação (em princípio, será implementado o algoritmo de [Hamasaki *et al.* 2007]), bem como, um segundo algoritmo de controle fundamentado na técnica *TF-IDF*.

Pretende-se realizar experimentos com três grupos: o grupo de teste, que utiliza a folksonomia produzida pelo algoritmo proposto neste trabalho; o grupo de controle I, que utiliza a folksonomia produzida pelo algoritmo de Hamasaki; e o grupo de controle II, que utiliza o conjunto de termos obtidos por *TF-IDF* a partir de um corpus. Cada grupo deve ser formado por pelo menos 10 pessoas. Os grupos de teste e de controle I e II utilizarão um conjunto de termos/*tags* como ponto de partida para modelagens conceituais de domínios variados. No caso do grupo de teste e do grupo de controle I, estes conjuntos de *tags* representam uma folksonomia em uma estrutura plana. Os algoritmos de teste e controle (I e II) utilizarão o mesmo conjunto de anotações como entrada. Os algoritmos de teste e controle I geram folksonomias por meio de suas heurísticas, enquanto que o algoritmo de controle II gera um conjunto de termos utilizando como corpus as *URLs* encontradas nas mesmas anotações. Os dados para induzir as folksonomias/gerar o conjunto de termos serão extraídos de aplicações sociais, tais como, *Delicious*® e *Bibsonomy*©.

Espera-se que o grupo de teste se depare com um número menor de divergências durante os experimentos de modelagem em relação aos grupos de controle. Portanto, pretende-se utilizar como métrica para efeito de comparação o número de divergências

geradas por cada grupo durante os experimentos. Para controlar as divergências geradas será utilizado o método *CoFolkconcept* [Hauagge *et al.* 2011]. O processo de modelagem no *CoFolkoncept* é colaborativo e se desenvolve da seguinte maneira: *i*) cada usuário constrói um modelo conceitual individualmente utilizando-se de um conjunto de *tags*/termos, produzindo, desta forma, um modelo conceitual particular; *ii*) os diferentes modelos conceituais de cada usuário são comparados a fim de se detectar divergências nas *tags*/termos escolhidos por cada usuário quanto ao tipo (conceito, instância ou relação) e à posição taxonômica (quando forem conceito ou instância); *iii*) resolvem-se as divergências por meio de discussões estruturadas de acordo com a metodologia *DILIGENT* [Tempich *et al.* 2005]; *iv*) gera-se uma nova versão do modelo conceitual que é consensual e repete-se o processo modificando-se individualmente o modelo consensual.

Serão realizados experimentos com diferentes parâmetros de geração das folksonomias a fim de determinar em quais condições o algoritmo de teste produz folksonomias que podem ser consideradas como conceitualizações compartilhadas em função do tipo de modelo conceitual almejado (se mais especializado ou menos especializado).

5. Contribuições da proposta

As contribuições desta proposta interessam aos pesquisadores que lidam com modelagem conceitual, em particular, com a atenuação do gargalo de aquisição de conhecimentos na modelagem conceitual, bem como no entendimento da utilização e dos limites de uso das folksonomias como fonte de informação na modelagem conceitual. Particularmente, propõe-se melhorar os algoritmos de indução de folksonomias pelo uso de autoria (autoridade cognitiva) e motivação das etiquetagens.

6. Agradecimentos

Agradecemos à Fundação Araucária pela bolsa de doutorado concedida a Josiane M. P. Ferreira durante o seu doutorado, no qual esta proposta será desenvolvida, e ao financiamento firmado no convênio 10/2011-FUP18520.

Referências

Angeletou, S., Sabou, M., Specia, L., Motta, E. (2007). "Bridging the Gap between Folksonomies and the Semantic Web: An Experience Report". In: *Proceedings of Bridging the Gap between Semantic Web and Web 2.0 Worshop, European Semantic Web Conference*.

Begelman, G., Keller, P., Smadja, F. (2006). "Automated Tag Clustering: Improving search and exploration in the tag space". In: *Proceedings of Collaborative Web Tagging Workshop at WWW'06*. Edinburgh, Scotland.

Cattuto, C., Benz, D., Hotho, A., Stumme, G. (2008). "Semantic Analysis of Tag Similarity Measures in Collaborative Systems". In: *Proceedings of 3rd Workshop on Ontology Learning and Population OLP3*, (pp. 39-43). Patras, Greece.

Feigenbaum, E. A. (1984). "Knowledge Engineering". *Annals of the New York Academy of Sciences*, 426: 91–107. doi: 10.1111/j.1749-6632.1984.tb16513.x

Guizzardi, G. (2005). “Ontological Foundations for Structural Conceptual Models”. Telematica Instituut Fundamental Research Series no. 15, Universal Press, The Netherlands, 2005, ISBN 90-75176-81-3.

Hamasaki, M., Matsuo, Y., Nishimura, T., Takeda, H. (2007). “Ontology Extraction using Social Network”, In *Proceeding of International Workshop on Semantic Web for Collaborative Knowledge Acquisition*, vol. 18700163, no. 18700163.

Hauagge, J. M., Tacla, C. A., Freddo, A. R., Molinari, A. H., Paraiso, E.C. (2011). “The Use of Well-founded Argumentation on the Conceptual Modeling of Collaborative Ontology Development”. In: *International Conference on Computer Supported Cooperative Work in Design* (CSCWD), 2011, Lausanne. Proceedings of the 2011 15th CSCWD. Piscataway: IEEE, 2011. v. 1. p. 113-119.

Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G. (2008). “Discovering shared conceptualizations in folksonomias”. In: *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, Issue 1, February 2008, p. 38-53.

Körner, C., Kern, R., Grahsl, H., Strohmaier, M. (2010). “Of Categorizers and Describers: An Evaluation of Quantitative Measures for Tagging Motivation”, In: *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pp. 157-166.

Maedche, A., Staab, S. (2001). “Ontology Learning for the Semantic Web”. IEEE Intelligent Systems, 16(2), 1-18. doi: 10.1109/5254.920602.

Mika, P. (2007). “Ontologies are us: A unified model of social networks and semantics”. In: *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1), 1-15. Springer. doi:10.1016/j.websem.2006.11.002

Robu, V., Halpin, H., Shepherd, H. (2009). “Emergence of consensus and shared vocabularies in collaborative tagging systems”. *ACM Transactions on the Web*, 3(4), 1-34. doi:10.1145/1594173.1594176

Schmitz, P. (2006). “Inducing ontology from Flickr tags”. In: *Proceedings of Collaborative Web Tagging Workshop, 15th WWW Conference*, Edinburgh.

Specia, L., Motta, E. (2007). “Integrating Folksonomies with the Semantic Web”. In: *4th European Semantic Web Conference* (Vol. 4519, pp. 624-639). Berlin Heidelberg, Germany: Springer-Verlag.

Strohmaier, M., Helic, D., Benz, D., Orner, C. K., and Kern, R. (2012). “Evaluation of Folksonomy Induction Algorithms”. To appear. *Transactions on Intelligent Systems and Technology*.

Tempich, C., Pinto, H. S., Sure, Y., Staab, S. (2005). “An argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT)”. *The Semantic Web: Research and Applications – Lecture Notes in Computer Science* (pp. 241-256). Springer.

Wilson P. (1983) Second-hand knowledge: An Inquiry into Cognitive Authority. WestPort: Greenwood Press.

Xu, X., Zhang, L., & Yu, Y. (2006). “Exploring social annotations for the semantic web”. In: *Proceedings of the 15th international conference on World Wide Web*, 417. New York, New York, USA: ACM Press. doi:10.1145/1135777.1135839

Abordagem para aquisição de conhecimento visual e refinamento de ontologias para domínios visuais*

Joel Luis Carbonera[1], **Mara Abel**[1], **Claiton M. S. Scherer**[2], **Ariane K. Bernardes**[2]

[1] Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

[2]Instituto de Geociências – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

{jlcarbonera,marabel}@inf.ufrgs.br, {claiton.scherer,ariane.kravczyk}@ufrgs.br

Abstract. *Em domínios visuais como a Medicina, a Meteorologia e a Geologia, as tarefas são realizadas através de uma aplicação intensiva do conhecimento visual dos especialistas. A natureza tácita do conhecimento visual impõe muitos desafios para Ciência da Computação em termos de aquisição, modelagem, representação e raciocínio. Neste trabalho será apresentada uma abordagem para aquisição de conhecimento visual e refinamento de ontologias em domínios visuais. Esta abordagem foi elaborada no contexto do projeto Obaitá, durante o processo de refinamento de uma ontologia de domínio para a Estratigrafia Sedimentar.*

1. Introdução

Domínios visuais são aqueles em que a resolução de problema inicia com um processo de casamento de padrões visuais, no qual são capturadas informações que serão utilizadas para suportar processos de inferência adicionais, em que são estabelecidas interpretações em um nível mais abstrato. Assim, domínios visuais são fortemente baseados na aplicação de conhecimento visual, que é 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]. Estas características dos domínios visuais levantam muitos desafios para a Ciência da Computação, devido à natureza tácita do conhecimento visual.

Este trabalho insere-se no contexto do projeto Obaitá [Lorenzatti et al. 2009, Carbonera et al. 2011, Torres et al. 2011], desenvolvido pelo grupo BDI (grupo de bancos de dados inteligentes da UFRGS). Este projeto tem como objetivo a investigação de abordagens integradas para aquisição, modelagem, representação e raciocínio sobre conhecimento visual. Esta investigação tem sido conduzida no domínio da Estratigrafia Sedimentar. Espera-se que este projeto tenha como um dos seus resultados uma ontologia de domínio para Estratigrafia Sedimentar, que viabilize a construção de diversos sistemas baseados em conhecimento, que operem sobre uma mesma conceitualização do domínio.

Nesta fase do projeto, partimos de uma ontologia de domínio em desenvolvimento [Lorenzatti et al. 2009], e investigamos como esta ontologia deveria ser refinada para

*As bolsas de estudo deste projeto foram financiadas pelo programa PETROBRAS PFRH 17. Outros materiais foram fornecidos pela ENDEEPER Knowledge Systems)

atender às demandas levantadas pela tarefa de *interpretação dos processos deposicionais*. Com isso, nosso foco imediato é o refinamento da ontologia para atender às demandas de sistemas baseados em conhecimento que viabilizem a descrição visual dos objetos de domínio (em função da conceitualização utilizada pela comunidade)[Abel et al. 2012] e permitam a realização computacional da tarefa de *interpretação visual* de processos deposicionais geradores das fácies sedimentares inspecionadas [Carbonera et al. 2011]. É importante salientar que nesta abordagem buscamos realizar processamento de *descrições simbólicas* de feições visuais, de modo que o foco não é o processamento de dados visuais brutos (como imagens). Assim, assumimos que os usuários dos sistemas realizam o processo de abstração das feições visuais percebidas em representações simbólicas, ajustadas à ontologia de domínio.

A tarefa de interpretação de processos deposicionais depende do *conhecimento das feições visuais* de interesse e do *conhecimento inferencial* que possibilita que o especialista estabeleça a relação entre tais feições visuais (tomadas como evidências) e a interpretação correspondente. Uma vez que uma parcela considerável deste conhecimento visual é *conhecimento tácito* [Polanyi 1966], há uma resistência por parte dos expecialistas em verbalizá-lo. Disto, segue-se que parte do conhecimento sequer é representado pela terminologia do domínio. Ou seja, parte dos fenômenos do domínio, presentes na conceitualização, podem não ter representações linguísticas. Com isso, as técnicas tradicionais para aquisição de conhecimento disponíveis na literatura (como *card sorting*, entrevistas estruturadas e não estruturadas, observação, limitação de informações, etc) não se mostram adequadas para realizar a aquisição de conhecimento visual [Abel 2001]. Este cenário motivou o desenvolvimento de uma abordagem de aquisição de conhecimento ajustada às necessidades deste projeto. Assumindo que a ontologia de domínio deve capturar o conhecimento necessário e suficiente para suportar a realização das tarefas de domínio, consideramos que a realização desta tarefa de interpretação pode oferecer uma boa ferramenta para avaliação da ontologia em desenvolvimento e, consequentemente, para revelar parcelas do conhecimento necessário faltante na ontologia em seu estado atual. Desta forma, utilizamos o próprio raciocínio especialista como uma ferramenta para atingir estes objetivos.

Na seção 2 será apresentado o domínio da Estratigrafia Sedimentar e seus principais objetos. Na seção 3 apresentaremos a nossa abordagem de aquisição de conhecimento visual e refinamento de ontologias. Na seção 4, apresentaremos um caso de uso desta abordagem o domínio. Na seção 5, apresentaremos algumas considerações finais.

2. Estratigrafia Sedimentar

A Estratigrafia Sedimentar é uma sub-área da Geologia que estuda as camadas que compõem a Terra e busca determinar a história da sua formação. Para alcançar este objetivo, o estudo inicia com a descrição visual de *corpos de rocha*. Estes, podem ser *testemunhos de sondagem*, que são cilindros de rocha retirados da subsuperfície terrestre por perfuração; ou *afloramentos*, que são exposições rochosas em superfície. A descrição destes corpos envolve discretizá-los em fácies sedimentares e descrever as caracteríticas visuais (atributo visual e valor) relevantes de cada uma delas. A *fácies sedimentar* é uma dada porção de um corpo de rocha, visualmente distinguível das porções adjacentes. No domínio, assume-se que cada fácies observada é o resultado de um determinado *processo deposicional*, que é um evento que envolve a interação complexa entre forças naturais e

sedimentos. A partir da informação visual contida na descrição de cada fácies o especialista oferece uma interpretação de um possível processo deposicional responsável por gerá-la. Em passos subsequentes, estes processos deposicionais interpretados são utilizados para reconstruir a história geológica que resultou na formação geológica analisada.

3. Abordagem para aquisição de conhecimento visual e refinamento de ontologias para domínios visuais

A abordagem aqui proposta foi sugerida por duas constatações:

- Uma análise de casos (descrições visuais de fácies associadas à interpretação do processo deposicional gerador correspondente) disponíveis na literatura revelou informações visuais que não poderiam ser descritas com base na ontologia em foco, em seu atual estágio de desenvolvimento. Isto sugeriu que a ontologia não oferecia uma representação suficiente da conceitualização do domínio.
- Quando o especialista observa diretamente uma fácies *in loco*, ele tem à disposição todo o conhecimento visual que potencialmente pode ser obtido a respeito da fácies. Isto permite que ele realize a interpretação do processo deposicional formador da fácies, do modo mais adequado possível. Por outro lado, quando delegamos a tarefa de interpretação para um procedimento computacional, os únicos recursos disponíveis para processamento são as descrições simbólicas das feições visuais da fácies, oferecidas pelo geólogo, representadas computacionalmente de acordo com a ontologia de domínio.

A partir destas constatações, consideramos a hipótese de que submeter o especialista às mesmas limitações impostas à máquina quando ela realizaria esta mesma tarefa, permitira avaliar a ontologia, bem como revelar uma parcela importante da conceitualização do domínio que eventualmente não estaria contemplada por ela. Tendo isto em mente, concebemos uma *abordagem para aquisição de conhecimento visual e refinamento de ontologias de domínio guiada por resolução de problemas em contextos de informações limitadas*. Em termos gerais, esta abordagem utiliza o raciocínio de resolução de problemas realizado pelo especialista como um meio indireto para avaliar a suficiência da conceitualização representada na ontologia, oportunizando a identificação de lacunas de conceitualização e a posterior eliciação desta conceitualização junto ao especialista. Na Figura 1 a abordagem é representada de forma esquemática. Parte-se de um conjunto de *casos selecionados* da literatura (a) resultante de um processo de *seleção de casos*, previamente realizado. Para cada caso (b), realiza-se a *tradução* (3) da descrição visual do objeto (*não estruturada*) (b) para uma versão *estruturada* (c) (5), em função da *ontologia de domínio* (d). É importante salientar que este processo de tradução pode não preservar toda a informação da descrição original, uma vez que a ontologia de domínio pode não suportar parte da conceitualização necessária. A *descrição estruturada* (c) é submetida então ao *especialista* (f), para que ele realize a *interpretação* (4) desta informação. Neste ponto, interpretação refere-se ao processo de raciocínio utilizado pelo especialista para resolver o problema em foco. O produto desta etapa é uma *interpretação do especialista* (e), realizada a partir das informações limitadas em (c). A seguir, identifica-se (5) a *interpretação do caso* (g), a partir do caso descrito de forma não estruturada (b). Então, verifica-se a *correspondência* (6) entre a *interpretação do especialista* (e) e a *interpretação original* (g). No caso de haver correspondência, o processo termina. No caso de não haver correspondência, há uma chance da discordância ter ocorrido devido

a uma insuficiência da ontologia em representar uma parcela da conceitualização, inviabilizando a estruturação de informações relevantes contidas no registro original. Por esta razão, busca-se *identificar* (7) *conhecimento faltante* (h) na *ontologia de domínio* (d). Isto pode ser feito perguntando-se ao especialista que tipo de informação seria necessária para suportar a interpretação original; ou apresentando a descrição original ao especialista, de modo que ele possa identificar algum conceito necessário para ajustar informaçoes que estão na descrição original, mas que não foram mantidos na descrição estruturada (devido a uma possível deficiência da ontologia). No caso de algum *conhecimento faltante* (h) ser identificado, ocorre o *refinamento* (8) da ontologia, através da incorporação deste novos conhecimento. O resultado do processo, ao fim da etapa (8), é uma ontologia de domínio refinada, em relação ao seu estágio inicial.

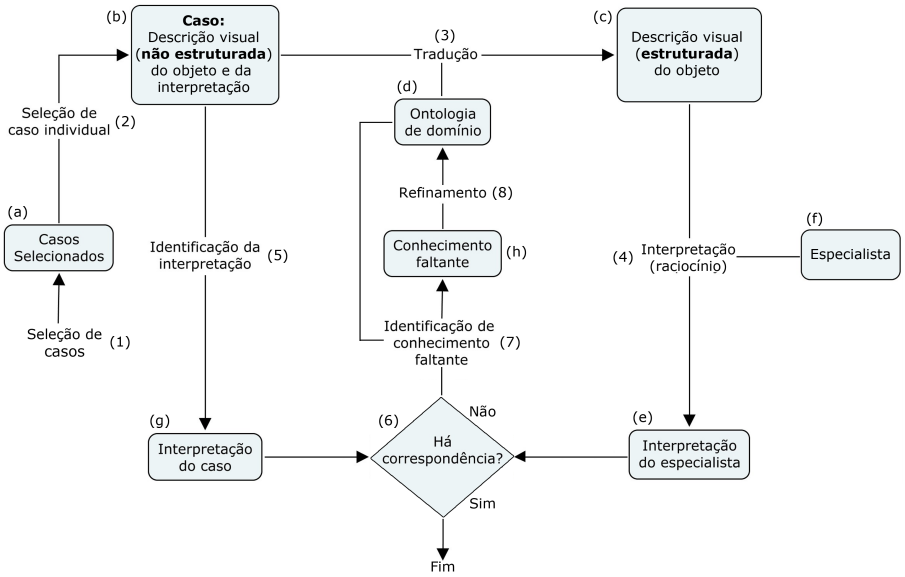


Figura 1. Representação esquemática da abordagem proposta

4. Aplicação da abordagem

Para realizar uma sessão de aplicação da abordagem com o especialista, primeiramente foram selecionados casos disponíveis na literatura, cada qual constituído por uma descrição das feições visuais de uma fácies sedimentar e de uma interpretação destas feições. Um exemplo de descrição selecionada da literatura é apresentado na Tabela 1, tal como ela está publicada (de forma não estruturada). Em um segundo passo, esta descrição foi traduzida para uma versão estruturada, usando os termos previstos pela ontologia em desenvolvimento, com o auxílio de dois geólogos em formação que conheciam a ontologia. Na Tabela 2 apresenta-se a versão estruturada. Neste estágio do desenvolvimento da ontologia, a versão traduzida contemplava apenas as informações que poderiam ser descritas com o atributo "Moda" e a "Estrutura Sedimentar", as demais informações da descrição foram consideradas "Observações adicionais", umas vez que não puderam ser descritas estruturadamente pela ontologia. Além destas informações, a Tabela 2 também apresenta a interpretação oferecida pela literatura, na coluna "Interpretação referencial", enquanto na coluna "Interpretação do especialista" é apresentada a interpretação que o especialista realizou, sem ter acesso às informações descritas na coluna "Observações adicionais". Na fase de verificação de correspondência, constatou-se uma diferença sutil entre

as interpretações. Na fase de verificação de conhecimento faltante, as informações rotuladas como "Observações adicionais" foram reveladas para o especialista, que constatou a relevância das informações não estruturadas para suportar a interpretação original (do caso selecionado). Dentre essas informações, a informação "Cascalho sustentado pelos clastos" revelou o atributo "Suporte de fábrica", do qual "Suportado por clastos" é um dos seus valores; enquanto a informação "clastos comumente imbricados" revelou o atributo "Orientação de Fábrica", do qual "Imbricado" é um dos valores. Uma vez revelados os atributos, o especialista listou outros valores possíveis para ambos. Com isso, realizamos o refinamento da ontologia, considerando estes dois novos atributos, e os seus respectivos valores possíveis. A Tabela 3 apresenta a mesma descrição de fácies, a partir da ontologia revisada. Comparando as Tabelas 1, 2 e 3, é possível notar que as informações tornam-se cada vez mais estruturadas ao longo das etapas, como consequência do refinamento da ontologia, através da incorporação de novos atributos e valores.

Descrição	**Interpretação**
Cascalhos sustentados pelos clastos, com estratificação horizontal pouco definida e clastos comumente imbricados	Barras longitudinais; depósitos residuais

Tabela 1. Exemplo de descrição de fácies encontrada na literatura. Descrição não estruturada

Moda	**Estrutura Sedimentar**	**Observações adicionais**	**Interpretação referencial**	**Interpretação do especialista**
Cascalho	Estratificação horizontal	Cascalho sustentado pelos clastos, clastos comumente imbricados, estratificação horizontal pouco definida	Barras longitudinais	Corrente trativa movendo cascalho, provavelmente barras longitudinais

Tabela 2. Exemplo de descrição de fácies submetida ao especialista. Estruturada de acordo com a ontologia em desenvolvimento

Moda	**Estrutura Sedimentar**	**Suporte de Fábrica**	**Orientação de Fábrica**
Cascalho	Estratificação horizontal	Suportado por clastos	Imbricada

Tabela 3. Exemplo de descrição de fácies estruturada a partir da ontologia revisada, com dois novos atributos para o conceito de fácies sedimentar

A utilização desta abordagem permitiu a identificação de 21 novos atributos com 62 novos valores. A comparação entre as versões da ontologia, antes e depois da aplicação desta técnica, fogem ao escopo deste trabalho, mas é apresentada em [Carbonera 2012];

5. Conclusão

Neste artigo apresentamos uma abordagem para aquisição de conhecimento visual e refinamento de ontologias para domínios visuais, bem como a aplicação desta abordagem no domínio da Estratigrafia Sedimentar. Esta abordagem apresentou bons resultados no que diz respeito à eliciação de novos atributos visuais descritivos. Este tipo de conhecimento é de fundamental importância em domínios visuais, visto que a tomada de decisão é fortemente dependente de caracterização visual qualificada dos objetos de domínio. Esta abordagem contribui para a aquisição de conhecimento e refinamento de ontologias porque, em geral, as abordagens disponíveis na literatura preocupa-nse com identificação

de outros tipos de conhecimento (conceitos, relações, taxonomias, etc), sem focar-se na identificação de atributos. Assim, a utilização desta abordagem em conjunto com outras, pode promover a aquisição de um espectro mais amplo de tipos de conhecimento.

A abordagem ainda não foi adequadamente formalizada, uma vez que ainda precisa ser aplicada em outros domínios visuais, com o intuito de verificar sua generalidade. É possivel que esta abordagem mostre-se adequada também em domínios não visuais. A verificação destes aspectos é um dos objetivos que pretendemos alcançar em trabalhos futuros. A abordagem apresentada pode ser considerada um resultado parcial do projeto Obaitá. É importante também salientar que a aplicação da abordagem pode revelar conceitos que não têm representação na terminologia do domínio, de modo que pode-se gerar termos que não são conhecidos pela grande comunidade da Geologia. No entanto, consideramos isto um aspecto positivo, no sentido de que pode-se revelar porções do conhecimento que ainda são difíceis de comunicar no domínio e sugerir termos que as representem. Estes termos revelados devem ser compreendidos como propostas, que devem ser avaliadas pela comunidade. O próximo passo natural é submeter a ontologia elaborada com a ajuda desta abordagem para a grande comunidade da geologia discuti-la em um processo de negociação de significados e construção colaborativa de ontologias [Torres et al. 2011].

Referências

Abel, M. (2001). *Estudo da Perícia em Petrografia Sedimentar e sua Importância para a Engenharia do Conhecimento*. PhD thesis, Universidade do Rio Grande do Sul (UFRGS).

Abel, M., Lorenzatti, A., Ros, L. F. D., da Silva, O. P., Bernardes, A., Goldberg, K., and Scherer, C. (2012). Lithologic logs in the tablet through ontology-based facies description. *AAPG Annual Convention and Exhibition*.

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.

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.

Polanyi, M. (1966). *The tacit dimension*. Anchor Day Books, New York.

Torres, G. M., Lorenzatti, A., Rey, V., da Rocha, R. P., and Abel, M. (2011). Collaborative construction of visual domain ontologies using metadata based on foundational ontologies. 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.

Towards an Ontological Process Modeling Approach

**Lucinéia Heloisa Thom, José Palazzo Moreira de Oliveira,
Jonas Bulegon Gassen, Mara Abel**

Departament of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil 91501-970
Email: lucineia/palazzo/jbgassen/marabel@inf.ufrgs.br

Abstract. *Process design requires a well understanding of the application domain. In practice, business analysts interview the domain experts and translate their understanding to process models. Often, the vocabulary used by the domain expert is very specific and difficult to understand by process analysts. Therefore, the process model elements can be named with inappropriate terms. Moreover, the lack of domain understanding by business analysts increases the probability of errors in the process model definition. Considering these aspects, this paper proposes an ontology process modeling approach. We expect that our approach helps to reduce the misunderstandings between the business analyst and the domain expert and at the same time it reduces the complexity and effort to build ontology and processes motivating the reuse of them. We have tested our approach in a case study regarding the Alzheimer domain.*

1. Introduction

The designing of particular processes from domains such as healthcare is very complex, not only because of their variety and need for flexibility, but also because they require the knowledge of very specific domain terms which can lead to interpretation problems, ambiguities and misunderstandings between the process analysts and the domain expert [Thom et al. 2006]. The medical knowledge management is a very acute concept, needed in different ways: to improve the patient care by a better know-how, to improve public health, to analyze comparative care process, to compare prescriptions. Related to these numerous needs, gathering medical information through ontologies for knowledge management can be very different as numerical documents make it easy.

Research on process design and ontologies have increased in recent years. One of the reasons is that ontologies and structured vocabularies in different domains help to make data understandable by machines. However, most of the existent approaches focus on building ontologies for the business process management domain as well as in the use of ontologies to add more semantics for the existent process model notations and execution languages. For example Haller and others [Haller et al. 2006] present an ontology that unifies both internal and external business processes, based on various existing reference models and languages from the workflow and choreography domain. The SUPER Project has developed the Business Process modeling Ontology (BPMO) [Norton et al. 2009]. Finally, the Unified Foundation Ontology was used to provide a common ontological foundation for goals, agents and business processes aiming to bridge the gap between these concepts [Guizzardi et al. 2010]. All these approaches use ontologies as a way to facilitate the understanding of the process domain. However, it remains a

need for more interdisciplinary approaches in the practice of process design and execution supported by ontology.

This paper proposes a methodology which altogether allows building the first notions towards an preliminary ontology using information from process models. We call it preliminary ontology because it must be completed with terms, attributes and relationships by a domain expert. Afterwards, this ontology can be used to support the design of new or to adapt existent process models to be reused. We expect that our approach reduces the misunderstandings between the business analyst and the domain expert facilitating the process design.

The remainder of this paper is organized as follows: Section 2 presents the core concepts about ontology used in this paper. Section 3 describes our proposed methodology to ontology building based on process models as well as to support the design of new process models. This Section also describes a case study we performed with students from the Ontology Course of a Public Brazilian University. We conclude with a summary and outlook in Section 4.

2. Background

An ontology defines a common vocabulary for researchers and other professionals who need to share information in a domain. It is a formal explicit description of concepts in a domain of discourse [Borst 1997]. Currently, there is no unique methodology to build an ontology. However, there exists a set of steps to ontology built described as follows: 1) determine the domain and scope of the ontology; 2) consider reusing existent ontologies; 3) enumerate important terms in the ontology; 4) define the classes, i.e. the representation of universal terms used to denote what is general in reality and the class hierarchy; 5) define the properties of the classes-slots (e.g. the class (concept), such as the *agitation* term in Fig. 2 can have the slot (property) such as *agitation level*); define the facets (values) of the slots (e.g. *hight/intermediate/down level of agitation*); create instances, that is to create instances of classes in the hierarchy.

Ontologies can be classified according to their level of generality [Guarino 1998]. *Top-level ontologies* describe very general concepts like space, object, event and action, which are independent of a particular domain; *Domain ontologies* and *task ontologies* describe, respectively, the vocabulary related to a generic domain (e.g. medicine) or a generic task or activity (e.g. diagnosing), by specializing the terms introduced in the top-level ontology; *Application ontologies* refers to a specific use or application focus, which the scope is specified through testable use cases. In particular, our case study presented in Section 4 refers to application ontology. We expect to (semi-)automatically merge several selected application ontologies to build a corresponding domain ontology. But it is a more complex task that we consider as future work.

3. PROCESS DESIGN SUPPORTED BY ONTOLOGY

In this section we propose a methodology which helps both process design and ontology building focusing on the alignment of both things. The methodology is composed of 5 basic steps (see Figure 1) i) understanding the purpose and domain of the process to be designed or reused; ii) designing a new process model from scratch or to adapt an existent one to be reused; iii) identifying basic terms from this process model in order to use these

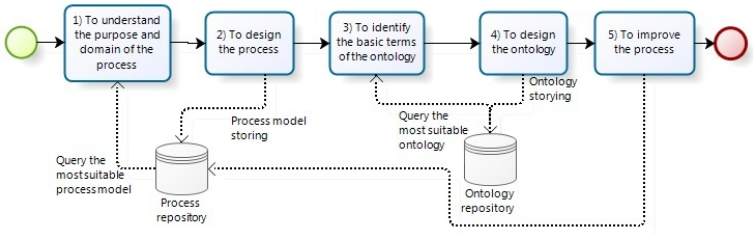


Figure 1. Metodology to process design using ontology.

terms as basic terms to build a new ontology or reuse an existent one; iv) building from scratch or to adapt an existent ontology to be reused and; v) improving a process model with information from the ontology.

Altogether, the methodology allows: i) designing a process model from scratch and a corresponding ontology; ii) reusing an ontology to support the design of a new process model and; (iii) to use an existent process model to support the building of a corresponding ontology. The steps of our methodology are described here and can be seen in Fig. 1 which must be read according to the Business Process Modeling Notation (BPMN).

In order to test the main steps of our methodology we developed a case study with students from the Ontology course of a public Brazilian University. The students were requested to design an *agitation behavior* state of a patient with the Alzheimer Disease and to build a preliminary ontology using our methodology.

1 **- Understanding the process purpose and domain.** In this phase the process analyst studies the application domain related to the business processes. To do that the analyst reads documents available in the organization and in the literature focusing in relevant information related to the process to be designed, i.e., trying to identify process fragments as well as activity roles. The idea is to reduce the number of interactions with the domain experts and use the interviews with them to complete the study and validate the initial findings. Based on the collected information, a query in the process repository is performed in order to verify whether there exist similar process models or fragments of them that can be reused, see Fig. 1, step 1). During the case study, the students performed an extensive study on the Alzeihmer Diseases. They had selected several health situations of an elderly suffering of Alzheimer and investigated how these situations could be described as process models.

2 **- Process design.** Based on the study performed in step 1 the business analyst can design a process model from scratch or simply adapt a process model obtained from the process repository. (cf. Fig.1, step 2). The design is generally completed with interviews with domain experts. If the query results in a process model, the process analyst decides whether the process: i) can be used without changes; ii) after it has been changed it will be saved in the process repository without be duplicated or; iii) after it has been changed it will be duplicated in the database. Otherwise, the process will be designed from scratch. In the case of option ii) or iii) be selected the repository will be updated. The *agitation behavior process* describes an elderly presenting disorientation, mood and behavior changes. Note

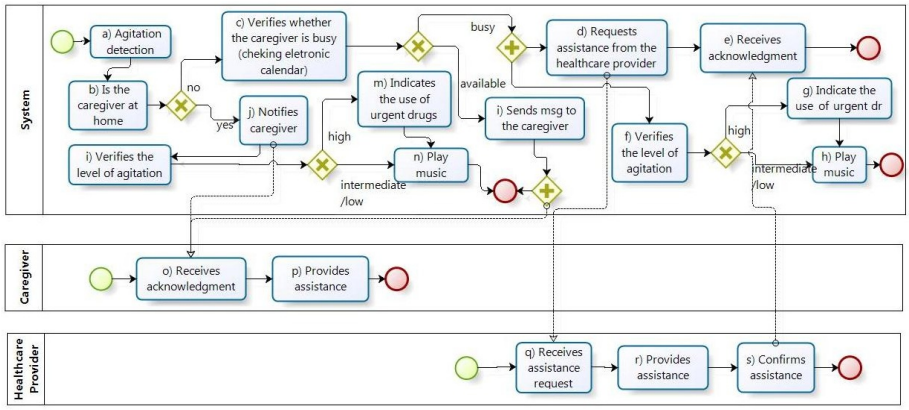


Figure 2. Agitation behavior process of a patient with Alzheimer Disease.

that this process is illustrative, which the students designed based on the literature and discussions we had during the ontology course (cf. Fig. 2). In real life, this process can present different behavior. As a result of this step, the students reported that they considered interesting to design situations which they thought could not be represented as process models. They learned how to focus on the core partial order of activities comprised by the studied scenarios.

3 **- Identification of basic terms of the ontology.** In this phase, the ontology designer or the process analyst will manually analyse the process and extract key terms considering the process elements such as the names of the activity labels and process participants. Based on selected key terms a query will be performed in the ontology repository to verify whether there exists a matching ontology. We are already investigating in the literature how to provide computational support for this task. Having the agitation process designed, the key terms related to the activities/roles of these processes as well as further co-related terms were identified and used to build the corresponding ontology. The extracted terms are underlined as shown in Fig. 3. This step was particulary interesting because the terms extracted from the process could be used as a reference to design the ontology. Therefore, the ontology includes the concepts that are relevant to the process model definition.

4 **- Preliminary Ontology design.** In this step, an ontology will be adapted or built from scratch (cf. Fig. 1, step 4). If there exists a matching ontology, the ontology designer decides: i) if the ontology can be reused without changes; ii) if the ontology will be adapted according with the process model or; iii) if a new ontology will be created from scratch. In case of adapting or creating a new ontology step 4 of our methodology will be performed, and the ontology repository must be updated. Otherwise, step 5 will be performed based on the ontology resulted from the query. When a new ontology is created, the terms extracted from the process are the initial step to build the ontology. These terms are the key terms of the preliminary ontology and of some relationships between the terms. The ontology designer will complete this ontology adding further terms, relationships, attributes and constrains. If necessary, inference rules will also be defined. In the case study, having the basic terms identified from the process models, the preliminary ontology was built. For that, additional terms not present in the process models but

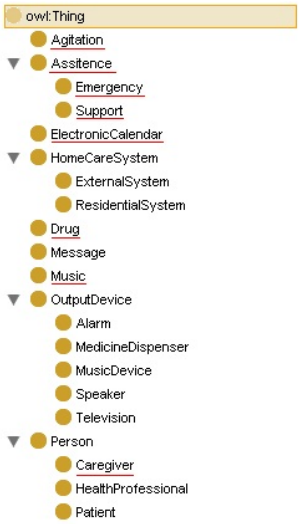


Figure 3. Alzheimer diseases ontology hierarchy.

related to the application domain were added, so that the ontology could be more complete. Moreover, some of the terms identified from the process were suppressed because they were not relevant for the ontology. The most relevant terms were the subject presented in the activity labels because they could be mapped to ontology classes. The next step was to build the classes of the ontology and for this task we have used the OWL (Web Ontology Language). As a result of this step the students reported that they considered easier to design the ontology using the process terms as first reference. They tried to understand the meaning of each terms and when necessary to relate further concepts from the domain.

5 - **Process model improvement.** The process model obtained from step 2 of our methodology can be reviewed with the assistance of the corresponding ontology. The process has then more suitable activity labels and names of participants using the improved ontology from step 4. So, the terms presented in the process including activity labels and activity participants can be reviewed. Moreover, we expect to integrate the ontologies with the process execution. In this work we have not really tested this part of our methodology because we are still working on that.

4. SUMMARY AND OUTLOOK

This paper showed that it appears to be suitable to use knowledge obtained from process models, at least as a starting point, to build the ontology and then to use the ontologies to support the design and execution of new process models within the same application domain.

Main advantages of our work can be summarized as follows: i) we propose a methodology to use information from process models to build a corresponding ontology and afterwards to use the ontology that must be completed by domain expertises to support the design of new similar process models. As a result, the ontology is more focused on information that are really used in the operational level of an organization (i.e., the process level); ii) we have tested the use of our methodology in a case study from the healthcare domain. The case study showed that the use of our methodology can reduce

the effort to understand the application domain at least regarding two scenario we have investigated; iii) based on several application ontologies we expected to develop a mechanism to (semi-)automatically obtain domain ontologies; iv) the ontology can be used to improve the process with more suitable terms concerning its domain. Currently, we are applying our methodology in a very interesting project regarding industrial robotic ontology. Our impressions lead to the conclusion that the robotic processes include components that will generate terms in the ontology that would not appear if the ontology would be directly build. Altogether, our approach covers several different situations that can occur: if we don't have neither the process model nor the ontology; if the process model does not exists yet, but we have the ontology to help creating it and; if we have an existent process model and want to extract an initial ontology.

As future work we intend to: i) perform additional case studies including applications from different domains; ii) develop a mechanism which allows to (semi-)automatically extract the key process terms used to build the ontology specially when dealing with large process models; iii) to investigate criteria to query the process and ontology repositories, including metrics to select the most suitable process and ontology resulted from a query; iv) to explore techniques to deal with the storing of processes and ontologies in the corresponding repositories (e.g., process and ontology similarities and the decision for replacement or duplication) and; vi) to develop mechanisms to (semi-)automatically build domain ontologies from application ontologies.

Acknowledgment

This research was partially supported by CNPq and CAPES, the first author works under a grant of the "Programa Nacional de Pós-Doutorado" (CAPES/PNPD).

References

Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Enschede.

Guarino, N. (1998). Formal ontology and information systems. pages 3–15. IOS Press.

Guizzardi, R., Guizzardi, G., Almeida, J. P. A., and Cardoso, E. C. S. (2010). Bridging the gap between goals, agents and business processes. In *Proceedings of the IV International i* Workshop (ISTAR 2010)*, pages 46–51, Hammamet, TunÃsia. 22nd International Conference on Advanced Information Systems Engineering, CEUR.

Haller, A., Oren, E., and Oren, A. H. E. (2006). A process ontology to represent semantics of different process and choreography meta-models. Technical report, Digital Enterprise Research Institute (DERI).

Norton, B., Cabral, L., and Nitzsche, J. (2009). Ontology-based translation of business process models. In *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, pages 481–486, Washington, DC, USA. IEEE Computer Society.

Thom, L. H., Reichert, M., Iochpe, C., and Moreira, J. P. (2006). Why rigid process management technology hampers computerized support of healthcare processes? In *Proceedings of the X Workshop on Medical Informatics*, pages 1522–1531, Belo Horizonte, Brazil. Computer Brazilian Society.

Ontologia dos Eventos Jurídicos: contribuições da semântica verbal

Carolina Müller[1], Rove Chishman[2]

[1]Doutoranda no Programa de Pós-graduação em Linguística Aplicada - UNISINOS

[2]Professora do Programa de Pós-graduação em Linguística Aplicada - UNISINOS

muller.carolina@ymail.com, rove@unisinos.br

Abstract. *This paper presents a preliminary study on the semantic description of the verbs of the Brazilian legal field. This is a study of the verbal semantics in order to construct a legal ontology of events. This work includes an exemplification of the analysis to be performed for the construction of ontology, presenting a ontology formalization proposal.*

Resumo. *Este artigo apresenta um estudo preliminar sobre a descrição semântica dos verbos do domínio jurídico brasileiro. Trata-se de um estudo acerca da semântica verbal com vistas à construção de uma ontologia dos eventos jurídicos. Este trabalho compreende uma exemplificação da análise a ser realizada para a construção da ontologia, apresentando uma proposta de formalização da ontologia.*

1. Considerações iniciais

Na área jurídica há um grande volume de documentos e informações relevantes armazenadas em diferentes bases de dados. Os documentos gerados por um processo jurídico podem servir de base para a produção de novos documentos, por esta razão a área jurídica necessita de ferramentas capazes de permitir a rápida recuperação da informação, possibilitando uma forma de apoio aos usuários na composição de novos documentos baseados na ampla base de dados constituída por sentenças, acórdãos, leis, etc.

Os sistemas de recuperação da informação são alvo de pesquisas cujo propósito está em tornar o conteúdo mais acessível. É este também o escopo do projeto **Tecnologias Semânticas e Sistemas de Recuperação de Informação Jurídica**[1], no qual esta pesquisa se insere. O grupo envolvido com este projeto tem como propósito "desenvolver e implementar um modelo semântico-conceitual do domínio jurídico brasileiro, de modo a ser integrado a sistemas de busca e recuperação de informação em sites que armazenam documentação jurídica."[2]

Uma ontologia do domínio jurídico possibilitará a ampliação da extração do conhecimento, permitindo o compartilhamento de informações relevantes aos usuários, sejam eles leigos ou especialistas. Através das ontologias as ferramentas de mineração de

[1] Projeto aprovado no Edital CAPES/CNJ Acadêmico 2010.
[2] Objetivo retirado do texto do próprio projeto [Chishman 2010]

dados podem ser aperfeiçoadas, permitindo maior exatidão no gerenciamento do conhecimento.

Para a construção da ontologia proposta, o grupo tomou como base as categorias recomendadas por Minghelli (2011), quais sejam: *eventos legais, instituições legais, documentos legais* e *participantes legais*. As diferentes categorias propostas serão estudadas separadamente em pesquisas em nível de mestrado e doutorado, a fim de compor uma única ontologia capaz de descrever semanticamente o domínio jurídico e possibilitar a recuperação da informação.

Neste artigo apresentamos reflexões preliminares acerca do domínio dos *eventos legais* visando à representação desta categoria na ontologia do direito brasileiro. Este estudo compreende a primeira etapa de análise semântica dos verbos jurídicos com vistas à construção da ontologia e faz parte de uma pesquisa ampla que comporá a tese de doutorado. Como se poderá constatar na próxima seção, a descrição que propomos para os eventos do domínio jurídico baseia-se em abordagens teóricas de cunho linguístico, amparando-se principalmente na Semântica Lexical e na Semântica de Frames.

2. A Semântica Verbal: abordagem teórica para a construção da ontologia

Consideramos que a semântica verbal pode ser analisada em três níveis: (a) aspectos lógico-semânticos, (b) aspectos gramaticais e (c) aspectos contextuais. Em relação aos aspectos lógico-semânticos avaliamos as relações lexicais de identidade e inclusão e as relações de exclusão e oposição – as relações paradigmáticas. Recorremos à Semântica de Frames e ao arcabouço da FrameNet [Fillmore et al. 2001] para abordar os eventos legais sob um viés contextual, considerando também as questões gramaticais relacionadas a situação e aos papéis temáticos – as relações sintagmáticas.

As relações paradigmáticas são associadas por Cruse (2000) com a coerência entre as classes, estando ligadas à identidade, inclusão, sobreposição e disjunção. As relações mais conhecidas no domínio lexical como sendo do eixo paradigmático são a hiponímia/hiperonímia[3], a meronímia/holonímia e a sinonímia/antonímia. Tais relações são fundamentais para a estruturação taxonômica da ontologia, sendo consideradas estruturantes para a organização das classes e subclasses.

Em trabalhos anteriores constatou-se que os aspectos lógico-semânticos não são suficientes para a descrição das entidades verbais, uma vez que estas ocorrem em um contexto e concorrem com outras entidades que complementam e/ou modificam seu significado [Müller 2011]. Assim, para tratar da semântica verbal há necessidade de adentrar no campo sintático-semântico e atentar para questões gramaticais e contextuais referentes ao significado.

[3] Miller e Fellbaum (1991) acreditam que as características que diferenciam dois verbos superordenados são diferentes das que diferenciam dois nomes; por esta razão denominam esta relação entre verbos de *troponímia*. No entanto, outros autores, como Cruse (2000) e Vossen (1997), não fazem tal distinção, apesar de considerarem as diferenças existentes entre uma taxonomia verbal e uma nominal, e mantêm a mesma nomenclatura.

Consideramos que os aspectos contextuais da semântica verbal são melhor descritos se considerada a abordagem teórica da Semântica de Frames, uma vez que os aspectos gramaticais e os papéis temáticos estão representados na estrutura dos *frames*[4]. A Semântica de Frames tem origem nos estudos de Fillmore (1982) e leva em conta fatores culturais e situacionais para descrever a estrutura cognitiva de um evento, ou seja, considera o chamado conhecimento enciclopédico, avaliando como o conhecimento geral do falante reflete na forma como ele interpreta o mundo e como compreende o significado das palavras.

Para Fillmore (1982), *frame semântico* é uma representação em forma de esquema de uma situação que envolve vários participantes, diversas propriedades e outros papéis conceituais onde cada uma das noções representa um *elemento de frame*, este, por sua vez, corresponde a uma categoria da FrameNet[5] (versão computacional a partir da Semântica de Frames). Assim, podemos concluir que cada argumento semântico relacionado a uma palavra corresponde a um *elemento de frame* do *frame semântico* ao qual a palavra está associada [Fillmore, Wooters e Baker 2001; Johnson e Fillmore 2000; Petruck 1996].

Na FrameNet um *frame* descreve uma situação típica de uma determinada língua, levando em consideração os aspectos culturais a ela relacionados e incluindo os participantes e suas condições. Cada *frame*, como uma categoria cognitiva, manifesta-se na língua por meio de palavras que o introduzem, isto é, *evocam* o *frame*, normalmente verbos.

Neste trabalho tomamos como fonte teórica a Semântica Lexical para tratar das relações taxonômicas que envolvem a estruturação da ontologia e a Semântica de Frames para tratar da descrição dos eventos jurídicos. Consideramos esta uma abordagem profícua para a organização da ontologia e apresentamos uma ilustração para a descrição do evento *julgamento* na seção seguinte.

3. Verbos do domínio jurídico: uma ilustração

Apresentamos uma ilustração preliminar de análise semântica relacionada aos verbos jurídicos e sustentada pelas abordagens teóricas discutidas anteriormente, com vistas a comprovar o potencial descritivo destas abordagens para a referida ontologia.

Nesta ilustração tomamos como exemplo o verbo JULGAR, escolhido por ser representativo do domínio jurídico. Realizamos uma busca no portal LexML[6] e encontramos 45550 ocorrências para o verbo *julgar* nas ementas das jurisprudências. Para a análise semântica do verbo *julgar* avaliamos os dez primeiros documentos

[4] Faz-se importante elucidar que o conceito de frame neste escopo difere do conceito difundido na área da Computação. Na área da Computação termo *frame* também é utilizado para referir-se à forma como dados se estruturam de modo a representar uma determinada situação, ou seja, compreendem um conjunto de informações sobre uma situação, que pode ser organizada através de propriedades (*slots*) que caracterizam cada circunstância [MINSKY 1974].

[5] Léxico computacional disponível em < https://framenet.icsi.berkeley.edu/fndrupal/>

[6] Portal especializado em informação jurídica e legislativa.

encontrados em nossa busca. Seguimos nossos pressupostos teóricos para identificar as relações paradigmáticas que envolvem este verbo e realizamos uma busca na FrameNet a fim de identificar os *frames* relacionados a esse verbo.

Após a análise semântica do verbo *julgar*, passamos à representação das classes e seus relacionamentos no editor de ontologias Protégé. Criamos a classe **Eventos** para acomodar os eventos jurídicos e a classe **Verbos** para inclusão dos verbos encontrados em nossa análise. Para a inclusão dos relacionamentos de sinonímia, hiperonímia, hiponímia, meronímia, criamos propriedades de objetos e relacionamos às respectivas classes (figura 1).

Conforme nossa análise *julgar* é sinônimo de *considerar*, *avaliar* e *fazer juízo*. Representamos a relação de sinonímia declarando um relacionamento de equivalência entre as classes, conforme pode ser visto no quadro pontilhado da figura 1.

Para as demais relações (hiponímia/hiperonímia e meronímia) criamos propriedades de objetos e estabelecemos os relacionamentos entre as classes que representam os diferentes verbos. Criamos a propriedade de objeto `éHipônimo` para declarar que *julgar* é hipônimo de *absolver*, *condenar*, *culpar*, *inocentar* e *sentenciar*. Para representar que *julgar* é parte de um *processo*, criamos a propriedade de objeto `éMerônimoDe`, conforme pode ser visualizado na figura 1. A propriedade de objeto `éHiperônimoDe` também foi criada para representar que os verbos *declarar* e *expressar* são hiperônimos de *julgar*. Criamos a propriedade `evocaEvento` para relacionar o verbo *julgar* ao evento **Julgamento**.

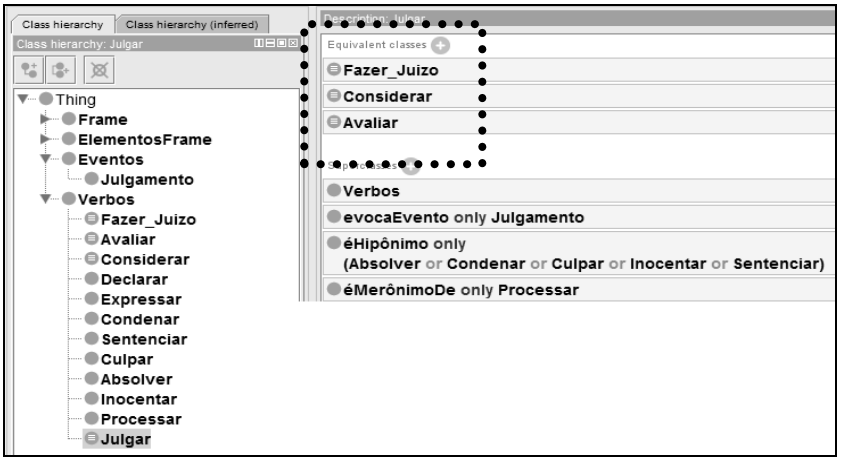


Figura 1: Representação das relações paradigmáticas. Fonte: elaborada pela autora

A representação das relações contextuais partiu dos frames existentes na FrameNet[7]. Encontramos o *frame* **Judgment** como representativo para o evento

[7] Salientamos que neste trabalho, a fim de ilustração, tomamos como apoio a FrameNet do inglês, porém estabelecemos as bases para os relacionamentos entre os verbos e os eventos jurídicos explicitados na JurFrameNetBr [Bertoldi 2011] que vem sendo desenvolvida em trabalho de pós-doutorado.

Julgamento. Para inclusão dos elementos nucleares[8] do *frame* criamos novas classes e subclasses: a classe **Frame** tem como subclasse **FrameJulgamento** e a classe **ElementosFrame** que tem como subclasses **Avaliador**, **Resultado**, **Avaliado**, **Expressão** e **Razões**. Para relacionar o evento **Julgamento** ao **FrameJulgamento** criamos a propriedade `evocaFrame`. Dessa forma determinamos que a classe **Julgar** evoca o evento **Julgamento** que está relacionado ao **FrameJulgamento**. No **FrameJulgamento** criamos uma propriedade para determinar que ele necessita ter todos os elementos de *frame*, isso foi estabelecido através da regra: `temElementosFrame` (figura 2).

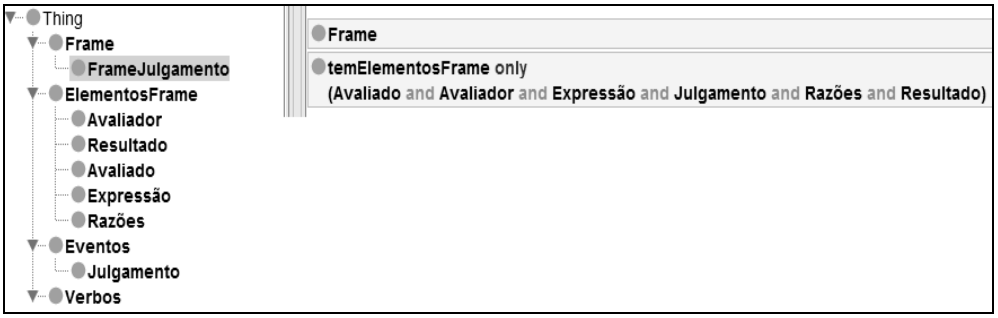


Figura 2: Representação das relações com *frames*. Fonte: elaborada pela autora

Através da estrutura de classes e das regras criadas estabelecemos a hierarquia da ontologia e uma forma de relacionamento com os *frames*, permitindo uma melhor descrição do verbo *julgar* no domínio jurídico.

4. Considerações finais

Nosso propósito neste artigo foi apresentar a fase inicial dos estudos para a construção de uma ontologia dos eventos jurídicos. Consideramos que a classe dos verbos compreende uma parte significativa para o estudo dos eventos jurídicos, fato este que nos leva buscar subsídios nas teorias semânticas que abarcam o significado verbal.

Buscamos referenciais que abarcassem três diferentes aspectos da semântica verbal: os lógico-semânticos, os gramaticais e os contextuais a fim da contemplar a amplitude do significado verbal e melhor descrever a amplitude da categoria *eventos jurídicos*.

Este ensaio nos mostra a Semântica de Frames como uma abordagem profícua para o estabelecimento de relações entre conceitos nas ontologias, permitindo um melhor detalhamento dos papéis de cada parte envolvida nos eventos jurídicos. Além de descrever os eventos, as classes correspondentes aos *elementos de frames* permitirão a ligação com as categorias *participantes legais*, *documentos legais* e *instituições legais*, formando a ontologia completa do direito brasileiro.

[8] Elementos de frame nucleares são aqueles que representam conceitos necessários para caracterizar um frame e manifestam-se na estrutura argumental evocada pelo predicador. Noção ligada à concepção de papéis semânticos que ocupam posições argumentais – elementos nucleares: papéis participantes e elementos não nucleares: papéis não-participantes.

Em nossas análises preliminares percebemos que o estudo dos eventos jurídicos também envolverá os nominais eventivos aos quais os verbos estarão relacionados, tais como *julgamento*, *absolvição*, *petição*, etc. Julgamos oportuno balizar que na descrição dos nominais eventivos também tomaremos como base teórica a Semântica de Frames, sendo que os *frames* para cada evento específico serão propostos e descritos.

A pesquisa de doutorado prevê, em suas etapas metodológicas, uma etapa linguística e uma etapa computacional, na qual as informações semânticas coletadas na análise serão inseridas no editor de ontologias Protégé, de modo a constituir a ontologia dos eventos do domínio jurídico brasileiro.

Referências

Cruse, D.A. (2000) "Meaning in Language: an Introduction to Semantics and Pragmatics". New York: Oxford University Press.

Bertoldi, A. (2011) "Semântica de Frames e recursos lexicais jurídicos: um estudo contrastivo". Tese de doutorado. São Leopoldo. UNISINOS.

Fillmore, Charles J. (1982) "Frame Semantics". In: *The Linguistic Society of Korea, Linguistic in the Morning Calm*, Seoul, Hanshin Publishing Co.

Fillmore, C., Wooters e Baker. (2001) "Building a Large Lexical Databank Which Provides Deep Semantics", In: *Proceedings of the Pacific Asian Conference on Language, information and computation* (PAVLIC 15), Hong Kong, 1-2.

Johnson, C., Fillmore, C. (2000) "The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure", In: *NAACL 2000 Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, Pages 56-62.

Miller, G. A. e Fellbaum, C. (1991) "Semantic Networks of English". In: *Levin e S. Pinker (eds), Lexical and Conceptual Semantics.* Cambridge, MA: Blackwell, p. 197-229.

Minghelli, T. D. (2011) "A relação de meronímia em uma ontologia jurídica", Dissertação de Mestrado, São Leopoldo, UNISINOS.

Minsky, M. A. (1974) "A Framework for Representing Knowledge". Artificial Intelligence Memo 306, MIT AI Lab.

Müller. (2011) "M_ONTO: Proposta de Modelagem Semântica para uma ontologia do domínio EAD". Dissertação de Mestrado. São Leopoldo. UNISINOS.

Petruck, M.R.L. (1996) "Frame semantics". In *J. Verschueren, J. Ostman, J. Blommaert, and C. Bulcaen, editors, Handbook of Pragmatics*. John Benjamins, Philadelphia.

Extração de Vocabulário Multilíngue a partir de Documentação de *Software*

Lucas Welter Hilgert, Renata Vieira, Rafael Prikladnicki

[1]Faculdade de Informática (FACIN) – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS – Brazil

Abstract. *This work aims for extracting multilingual vocabulary from software documentation in order to create resources for improving machine translation assisted comunication in the context of software requirement meetings involving multilingual teams. The objective of this paper is to present the initial results obtained related to the research data (corpus) construction.*

Resumo. *Este trabalho tem por objetivo a extração de vocabulário multilíngue a partir de documentação de software, visando construir recursos para o melhoramento da comunicação assistida por tecnologias de tradução de máquina no contexto de reuniões de requisitos de software envolvendo times multilíngues. Este artigo tem por objetivo apresentar os resultados iniciais obtidos durante a construção do material de pesquisa (corpus).*

1. Introdução

O trabalho aqui apresentado encontra-se inserido no âmbito do projeto "*O Efeito do Processamento da Linguagem Natural no Desenvolvimento da Capacidade do Brasil no Mercado Global de Desenvolvimento de Software*", cujo principal objetivo é auxiliar na inclusão de equipes brasileiras no mercado global de desenvolvimento, mediante a investigação e utilização de métodos, técnicas e ferramentas da área de Processamento da Linguagem Natural (PLN).

Dentre as tecnologias de PLN, dá-se enfoque especial aos serviços de tradução de máquina, considerados como uma solução alternativa para as dificuldades linguísticas (diferentes idiomas) encontradas durante reuniões de equipes multilíngues de desenvolvimento de *software* [Calefato et al. 2012] [Yamashita and Ishida 2006].

Como apresentado em diferentes trabalhos [Calefato et al. 2011] [Calefato et al. 2012] [Yamashita and Ishida 2006], as tecnologias de tradução automática ainda estão longe da perfeição, possuindo uma série de questões a serem resolvidas, para as quais, uma das possíveis soluções é a construção de vocabulários multilíngues específicos do domínio [Nakatsuka et al. 2010].

Sendo assim, este trabalho tem como principal objetivo a construção de um vocabulário multilíngue referente às práticas de desenvolvimento distribuído de *software*, com a finalidade de auxiliar os serviços de tradução de máquina empregados durante as reuniões das equipes.

2. Contextualização do Trabalho

Dos trabalhos referenciados, destaca-se o experimento conduzido por Calefato *et al.* [Calefato et al. 2012], executado em uma parceria entre pesquisadores brasileiros (PUCRS) e italianos (Universidade de Bari), cujos registros (*logs*) foram utilizados como principal fonte para a investigação de problemas relacionados à tradução automática aplicada ao contexto de tarefas colaborativas.

Neste experimento, equipes multilíngues (formadas por 2 participantes brasileiros e 2 italianos) executaram, colaborativamente, tarefas relacionadas à engenharia de requisitos, utilizando, de forma alternada, o inglês como idioma comum, e seus idiomas nativos em conjunto com serviços de tradução de máquina.

A partir da análise dos registros do experimento, diferentes tipos de problemas foram encontrados sendo que, neste trabalho, optou-se por priorizar aqueles relacionados ao vocabulário, destacando-se: (1) traduções inconsistentes, (2) abreviações de termos, (3) erros de digitação.

Como exemplo de tradução inconsistente, pode-se mencionar o termo *"release"* , traduzido de diferentes formas (*"entrega"* e *"lançamento"*, por exemplo) para os mesmos contextos (motivo da inconsistência), ou mesmo mantido como *"release"*. Este tipo de inconsistência, pode induzir a problemas de compreensão entre os participantes da reunião [Nakatsuka et al. 2010].

A abreviação de termos se demonstrou um problema, principalmente quando aplicada à termos cujas abreviações possuem significado próprio. Como exemplos deste tipo de problemas, pode-se mencionar *"bluetooth"*, simplificado como *"blue"* gerando a tradução (inadequada ao contexto) *"azul"*, e *"ring tone"*, abreviado como *"ring"* e traduzido como *"anel"*.

Uma das possíveis soluções encontradas é a utilização de funcionalidades de autocomplementação para auxiliar os participantes durante a escrita. Estas funcionalidades podem ser alimentadas com um vocabulário inicial, a ser ampliado no decorrer da comunicação.

Em relação aos erros ortográficos, pode-se mencionar o termo *"bluetooth"* para o qual foram encontradas 5 diferentes grafias sendo 4 destas incorretas (*"blutooth"*, *"bluetoth"*, *"bluetooh"* e *"blutoofh"*).

Este tipo de erro é frequentemente tratado através de funcionalidades de correção ortográfica (*spellchecking*) sendo que essas dependem da existência de um vocabulário para a identificação das formas corretas das palavras.

Por fim, durante o levantamento dos recursos utilizados como base para a construção de um vocabulário multilíngue, identificou-se a necessidade desse tipo de recurso na construção automática de corpus, tarefa esta que pode ser empregada para ampliação do corpus existente.

3. Construção de Vocabulário Multilíngue

O processo de extração de vocabulário multilíngue, utilizado neste trabalho, encontra-se demonstrado, de forma simplificada, na Figura 1. Esse consiste em, dado um corpus multilíngue, realizar a extração do vocabulário equivalente aos idiomas envolvidos e,

posteriormente, avaliá-lo [Ha et al. 2008] [Daille and Morin 2005]. Uma descrição mais detalhada dessas etapas será apresentada a seguir.

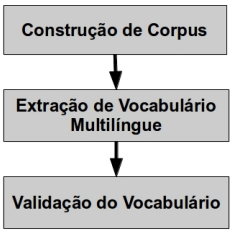


Figura 1. Processo genérico de extração

3.1. Construção do Corpus

Para a construção do corpus multilíngue foram considerados: a) textos paralelos (textos acompanhados por suas respectivas traduções [Ha et al. 2008]) e b)textos comparáveis (textos em diferentes idiomas que compartilham características comuns como tema, por exemplo [Daille and Morin 2005]).

Como fonte principal de material, optou-se pela utilização da documentação de *softwares open source* (código aberto), devido a sua disponibilidade para diferentes idiomas e ao tipo de licenciamento utilizado por esses. Posteriormente, foram incluídos à lista livros da Engenharia de *Software*.

Os materiais anteriormente mencionados foram coletados a partir das páginas oficiais dos projetos. Versões multilíngues do manual de usuário da ferramenta de versionamento TurtoiseSVN, por exemplo, foram extraídos a partir do sítio *"http://tortoisesvn.net/support.html"*. Como uma das colaborações deste trabalho, o *corpus* construído, relacionado a projetos *open source*, será futuramente disponibilizado em um repositório.

Conhecidas as fontes, a coleta dos documentos foi conduzida de forma manual. Abordagens automáticas para a construção do corpus foram cogitadas, no entanto, devido à falta de termos iniciais (*seeds*) e a complexidade associada a avaliação dos textos coletados por essas, optou-se por não utilizá-las (pelo menos inicialmente).

3.2. Extração de Vocabulário Multilíngue

O processo de extração de vocabulário multilíngue consiste na obtenção de equivalências entre palavras de diferentes idiomas. A execução desse varia de acordo com o tipo de corpus empregado (paralelo ou comparável).

Dado o trecho de sentença "...será a nova versão do *software*...", por exemplo, e seu trecho equivalente nos documentos em inglês "*...will be the new release of the software*", o método de extração deve estabelecer a relação entre as palavras *"release"* e *"versão"*.

Em relação a corpus paralelo, um dos procedimentos mais utilizados é o alinhamento textual, sentencial e/ou lexical, sendo que esse consiste na identificação de trechos correspondentes entre textos considerados como paralelos (texto e sua respectiva tradução)[Ha et al. 2008].

Quanto a corpus comparável, o estabelecimento de equivalências costuma ser realizado mediante a utilização de vetores de contexto (que levam em consideração as palavras próximas ao termo a ser traduzido) em conjunto com dicionários multilíngues de domínio geral [Daille and Morin 2005].

O processo de extração de vocabulário empregado até momento, baseia-se na extração de *n*gramas, e é realizado de acordo com os seguintes passos:

1. Os documentos extraídos são normalizados quanto a seu formato, sendo convertidos para documentos de texto sem formatação (*plain text*);
2. Os textos são separados em sentenças (*Sentence detection*) que, por sua vez, são separadas em seus símbolos formadores (palavras, sinais de pontuação, etc.);
3. Numerais, sinais de pontuação, símbolos especiais (marcadores, por exemplo) e palavras muito comuns do idioma (*stopwords*) são removidos;
4. As palavras restantes são submetidas a um processo de lematização para a obtenção de suas respectivas formas canônicas;
5. Listas de *n*gramas (sequências *n* palavras) são construídas e posteriormente contabilizadas. Neste trabalho foram considerados unigramas, bigramas e trigramas;

Com a exceção da lematização dos textos em português (conduzida com o lematizador da ferramenta CoGroo [Kinoshita et al. 2007]), as demais etapas foram realizadas com o conjunto de ferramentas disponibilizadas pelo NLTK [Bird et al. 2009].

3.3. Avaliação do Vocabulário

A avaliação do vocabulário construído pode ser realizada tanto de forma manual quanto automática. A validação manual consiste na revisão do vocabulário por um tradutor profissional ou por um especialista da área, enquanto na validação automática o vocabulário construído é comparado com um padrão de referência (*golden standard*) previamente criado [Daille and Morin 2005] [Ha et al. 2008].

4. Resultados Parciais

Até o momento, como recurso selecionado para pesquisa, tem-se um corpus multilíngue composto por 567.458 palavras (unigramas) em inglês e 331.626 palavras em português.

Exemplos de palavras (unigramas) extraídas a partir dos textos em português do corpus, mediante o processo apresentado na seção 3.2, são: “Tela”, “Contato”, “lista”, “agenda”, *“wi-fi”*, *“bluetooth”*, “calculadora” e *“sms”*.

Nesses, são encontrados tanto termos de domínio como *“Contato”* (dispositivos móveis), quanto termos que designam tecnologias como *“bluetooth”*, por exemplo. A lista de palavras (assim como a de bigramas e trigramas) será melhor investigada em busca de palavras e termos comuns a diferentes domínios, com enfoque principal na terminologia de Engenharia de *Software*.

Quanto aos materiais auxiliares, buscou-se por vocabulários já compilados, relevantes ao domínio. A Tabela 1 demonstra os principais vocabulários encontrados juntamente com a quantidade de termos contidos em cada um desses. Vale ressaltar que apesar desses serem monolíngues (inglês), são compostos por termos diretamente relacionados ao domínio, sendo que traduções para os mesmos serão buscadas.

Vocabulário	Termos
System and Software Engineering–Vocabulary [ISO/IEC/IEEE 2010]	3.349
Standard Glossay of Software Engineering Terminology [IEEE 1990]	1.300
Glossário "*Software Engineering*" [Sommerville 2010]	167
Lista de Assuntos "*Software Engineering*" [Sommerville 2010]	1.600

Tabela 1. Vocabulários obtidos

Por fim, foram obtidos registros (*logs*) de comunicação entre equipes de desenvolvedores. A partir dos registros do experimento de Calefato *et al.* [Calefato et al. 2012] foram extraídas mensagens em português (449), italiano (694) e inglês (874). Posteriormente, foram obtidos registros de comunicação entre desenvolvedores da fundação *Mozilla* compostos por 161.316 mensagens (aproximadamente 1.669.000 palavras).

4.1. Aplicabilidade dos Recursos

Uma análise inicial dos recursos levantados demonstrou a aplicabilidade desses na solução de problemas identificados durante a análise dos registros (seção 2).

Em relação aos problemas de tradução inconsistente, partindo do termo *"release"*, previamente apresentado, buscou-se nos textos em inglês do corpus por ocorrências desse. Dentre os contextos nos quais o termo foi encontrado, 6 foram selecionados, sendo que seus trechos equivalentes foram buscados nos textos em português. Para os 6 contextos avaliados o termo foi coerentemente traduzido como *"versão"*, indicando uma tendência na utilização desta tradução no domínio em questão.

Referente aos problemas de abreviação e erros ortográficos, para ambos os exemplos apresentados (*"bluetooth"* e *"ring tone"*), os termos foram encontrados no material compilado, indicando que estes poderiam ser utilizados em funções de correção ortográfica (erros ortográficos) e auto-complementação (abreviação), ambos recursos de auxílio a escrita.

Mais exemplos de aplicabilidade do material compilado na solução dos problemas mencionados serão obtidos durante a execução das próximas etapas do processo de extração.

O vocabulário bilíngue extraído pode vir a ser empregado, ainda, na tradução de ontologias existentes na área de Engenharia de *Software*. Conhecida a natureza multilíngue do desenvolvimento global de *software*, torna-se importante que estas ontologias encontrem-se disponíveis em mais de um idioma.

5. Conclusões

A utilização de serviços de tradução simultânea de máquina, considerada como uma solução alternativa ao inglês durante reuniões de equipes distribuídas, tem seu desempenho comprometido devido à problemas de tradução de máquina, dentre os quais destacam-se os apresentados na seção 2.

Entre os diferentes tipos de problemas observados, optou-se por priorizar aqueles relacionados ao vocabulário, para os quais a solução proposta consiste na construção de um vocabulário multilíngue das práticas usuais do processo de desenvolvimento de *software*.

No entanto, como apresentado na seção 3.2, a construção de um vocabulário multilíngue depende da existência de um corpus multilíngue previamente compilado. Neste trabalho, identificamos os recursos disponíveis para a construção deste vocabulário.

Em relação à primeira etapa deste trabalho, o principal resultado obtido foi um conjunto de materiais formado por um corpus multilíngue, composto por manuais de *software*, conjuntos de vocabulários referentes ao domínio, e registros (*logs*) de comunicação entre desenvolvedores.

Uma vez o corpus compilado, as próximas etapas consistem na aplicação de métodos de extração de vocabulário multilíngue (baseados em *corpus* paralelo), sobre o *corpus* construído, seguida da avaliação manual do vocabulário extraído.

Referências

Bird, S., Loper, E., and E., K. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc.

Calefato, F., Lanubile, F., Conte, T., and Prikladnicki, R. (2012). Assessing the impact of real-time machine translation on requirements meetings: A replicated experiment. In *6th Int'l Symposium on Empirical Software Engineering and Measurement (ESEM'12) (to appear)*, page 19–20.

Calefato, F., Lanubile, F., and Prikladnicki, R. (2011). A controlled experiment on the effects of machine translation in multilingual requirements meetings. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*, pages 94 –102.

Daille, B. and Morin, E. (2005). French-english terminology extraction from comparable corpora. In Dale, R., Wong, K.-F., Su, J., and Kwong, O. Y., editors, *IJCNLP*, volume 3651 of *Lecture Notes in Computer Science*, pages 707–718. Springer.

Ha, L. A., Fernandez, G., Mitkov, R., and Pastor, G. C. (2008). Mutual bilingual terminology extraction. In *LREC*. European Language Resources Association.

IEEE (1990). Ieee standard glossary of software engineering terminology std 610.12-1990.

ISO/IEC/IEEE (2010). Systems and software engineering – vocabulary.

Kinoshita, J., Salvador, L. N., and Menezes, C. E. D. (2007). Cogroo - an openoffice grammar checker. In *Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, ISDA '07, pages 525–530, Washington, DC, USA. IEEE Computer Society.

Nakatsuka, M., Yasunaga, S., and Kuwabara, K. (2010). Extending a multilingual chat application: Towards collaborative language resource building. In *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pages 137 –142.

Sommerville, I. (2010). *Software Engineering*. Addison-Wesley, Harlow, England, 9. edition.

Yamashita, N. and Ishida, T. (2006). Effects of machine translation on collaborative work. *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work CSCW 06*, page 515.

Construção de Modelos Conceituais a Partir de Textos com Apoio de Tipos Semânticos

Felipe Leão, Thaíssa Diirr, Fernanda Baião, Kate Revoredo

NP2Tec – Núcleo de Pesquisa e Prática em Tecnologia
Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur, 296 Urca – Rio de Janeiro – Brasil

{felipe.leao, thaissa.medeiros, fernanda.baiao, katerevoredo}@uniriotec.br

Abstract. *The conceptual modeling process involves the understanding of domain concepts and its representation in diagrams. The knowledge about the domain can be obtained from various sources, most of them being based on natural language. Therefore, the automatically creation of conceptual models becomes interesting. This paper presents an analysis of the feasibility of automating a linguistic approach with semantic focus to allow the construction of ontologies from natural language texts.*

Resumo. *O processo de modelagem conceitual envolve a compreensão de conceitos do domínio em questão e sua posterior representação em diagramas. O conhecimento sobre o domínio pode ser obtido a partir de diversas fontes, sendo a maior parte delas baseadas em linguagem natural. Portanto, a criação de modelos conceituais de forma automática se mostra interessante. Este trabalho apresenta uma análise da viabilidade de se automatizar uma abordagem linguística com foco semântico capaz permitir a construção de ontologias a partir de textos em linguagem natural.*

1. Introdução

A modelagem conceitual constitui uma das principais fases na construção de um sistema de informação. Em essência, tal fase busca definir os conceitos envolvidos na descrição do problema, sendo o seu resultado um substrato de grande valor para a boa compreensão geral dos requisitos e auxílio para as etapas seguintes na modelagem de qualquer solução computacional. O processo envolve a compreensão de conceitos relacionados ao domínio em questão e a posterior representação desses conceitos em modelos, como por exemplo ontologias.

Por outro lado, a transmissão de conhecimento do especialista no domínio para o modelador durante a construção de uma ontologia é geralmente feita através da troca de informações em textos (transcrições de entrevistas, documentos regulatórios, entre outros). Para obter uma representação fiel do domínio modelado com qualidade semântica, é necessário, além de um metamodelo representativo, que o modelador interprete corretamente os conceitos do domínio e não influencie sua compreensão com experiências e conhecimentos anteriores [Castro *et al*., 2011].

Diversos trabalhos propuseram abordagens para modelagem conceitual de dados através de ontologias com base em conceitos linguísticos, entre eles Castro *et al*. (2011) propôs um método baseado em linguística e foco semântico. O processo é naturalmente

composto por duas atividades principais, (i) a aquisição de conceitos sendo utilizados no domínio a ser modelado e (ii) a representação de tais conceitos através da linguagem de modelagem OntoUML [Guizzardi, 2011]. Apesar de esse método auxiliar a produção de ontologias como modelos conceituais de dados com qualidade semântica, a falta de suporte automatizado é um ponto negativo, uma vez que as definições formais de uma língua podem ser de grande complexidade. É interessante então buscar meios que viabilizem a automatização da abordagem.

Este trabalho se propõe a analisar a aplicação de técnicas automatizadas para reconhecimento sintático e semântico à abordagem de Castro et al. (2011), auxiliando e agilizando a realização dos passos que a compõem. Técnicas utilizadas atualmente para realização de processamento de linguagem natural como *POS Tagging* e *Semantic Tagging* foram pesquisadas.

O restante do trabalho está organizado da seguinte forma: A seção 2 apresenta a abordagem linguística para modelagem com foco semântico, incluindo a linguagem OntoUML. A seção 3 descreve as técnicas analisadas e as duas abordagens consideradas. A seção 4 expõe considerações finais sobre o trabalho e trabalhos futuros.

2. Abordagem Linguística para Modelagem Conceitual com Foco Semântico

Castro et al. (2011) propôs uma abordagem linguística para modelagem conceitual que parte da linguagem natural para construção de ontologias em OntoUML. Esta seção explicita os passos propostos para obtenção dos modelos, além da própria linguagem OntoUML.

2.1. OntoUML

A OntoUML é uma linguagem de modelagem conceitual ontologicamente bem fundamentada, que pode apoiar a fase de análise de domínio na engenharia de sistemas de informação. Ela foi proposta como um perfil da UML 2.0 e incorpora axiomas e distinções ontológicas presentes na ontologia de fundamentação UFO (*Unified Foundation Ontology*) [Guizzardi, 2011].

Os elementos da UFO podem ser *Universal* (tipos que determinam coleções de conceitos que se referem a coisas ou seres) ou *Individual* (instâncias de tais coleções). Essas classes se subdividem em outras que, por sua vez, também possuem sub-hierarquias [Guizzardi, 2005 apud Castro et al., 2011]. Na hierarquia de *Universal*, por exemplo, poderíamos citar a classe *Substantial Universal* que corresponde a propriedades intrínsecas (não relacionais) que determinam classes ou coleções de seres ou coisas materiais e mantêm sua identidade, mesmo passando por mudanças.

As classes da linguagem de modelagem conceitual OntoUML são especializações das classes abstratas da UFO, herdando meta-propriedades e/ou restrições. Alguns desses construtos são [Guizzardi, 2005; Guizzard, 2011; e Castro et al., 2010]:

- *Kind* (tipo): seres complexos ou coisas relacionalmente independentes claramente identificados. (e.g. animais, plantas, time de futebol).
- *Quantity* (quantidade): classes de substâncias de massa (e.g. água).

- *Collective* (coletivo): coleções ou coisas vistas e percebidas como uma estrutura uniforme (e.g. floresta, baralho de cartas).
- *Phase* (fase): estágios ou fases na existência de um ser (e.g. lagarta e borboleta são partições - *phase* - de um lepidópteros - *kind*).

2.2. Modelagem Conceitual com Foco Semântico

O processo de modelagem de dados conceitual é similar a uma atividade de tradução, pois consiste em entender os conceitos representados em uma linguagem natural e, então, representar esses mesmos conceitos em uma linguagem de modelagem [Castro *et al*., 2011]. Para realizar essa tradução é preciso compreender e comparar as linguagens, a fim de elaborar modelos conceituais (ontologias) semanticamente precisos que apresentem o mesmo significado do texto em linguagem natural [Castro *et al*., 2010 e 2011].

O vocabulário de qualquer linguagem natural é dividido em classes de palavras que podem ser fechadas ou abertas. Classes abertas são as que carregam carga semântica (substantivos, adjetivos, verbos e advérbios) e, por isso, são o foco da modelagem conceitual. As classes de palavras podem ainda ser especializadas em tipos semânticos propostos por Dixon (2005). Alguns destes tipos especializam os substantivos de referência concreta, que têm maior importância para modelagem conceitual por nomearem seres e coisas. São eles:

- *Animate* (animados): abrangem os animais não humanos;
- *Human* (humanos): referentes a seres humanos. São subdivididos em *Kin* (e.g. filho, pai), *Rank* (e.g. goleiro, professor) e *Social group* (e.g. companhia).
- *Parts* (partes): são partes de outras coisas ou seres, incluindo as partes corpóreas.
- *Inanimate* (inanimados): subdivididos em *Artifacts* (e.g. livro), *Flora* (e.g. árvore), *Celestial and Weather* (e.g. lua, vento) e *Environment* (e.g. água).

A partir do entendimento das propriedades das linguagens naturais e da OntoUML, Castro *et al*. (2010) definiram mapeamentos de tipos semânticos relacionados a substantivos, verbos e adjetivos para construtos da OntoUML. A Tabela 1 apresenta os construtos da OntoUML relacionados aos tipos semânticos de substantivos concretos.

Tabela 1 - Mapeamento de Tipos Semânticos de Substantivos Concretos [Dixon, 2005] para OntoUML. Fonte: Castro et al (2010)

Tipos Semânticos	Construtos OntoUML
Animate, Human, parts, Inanimate, Inanimate/Artifacts, Inanimate/Celestial and Weather, Inanimate/ Flora	Kind
Human/Social Group	Collective ou Kind
Inanimate/Environment	Quantity
*Vários	Phase

Esses mapeamentos são utilizados na abordagem de modelagem conceitual proposta em Castro *et al*. [2010, 2011]. A abordagem parte de textos descritivos sobre o universo a ser modelado, produzidos pelos especialistas do domínio. Os passos que

compõem a abordagem estão listados a seguir [Castro *et al*., 2011], onde o mapeamento da Tabela 1 é aplicado no passo 5.

1. *Decomposição do texto em sentenças simples* (sentenças não interrogativas, na voz ativa e contendo somente uma oração);
2. *Levantamento e esclarecimento de dúvidas junto ao especialista do domínio;*
3. *Identificação dos signos do universo modelado* (sujeitos, verbos e objetos);
4. *Associação entre os signos identificados e os tipos semânticos correspondentes;*
5. *Mapeamento entre os tipos semânticos e os construtos da OntoUML;*
6. *Criação do modelo*.

3. Técnicas para Reconhecimento Sintático e Semântico

É possível observar que os passos 1, 3 e 5 da abordagem de Castro et al (2010) são diretamente automatizáveis, através de técnicas como *Tokenizing* e *Parsing* e os mapeamentos de tipos semânticos para OntoUML. O passo 2 consiste em uma tarefa não automatizável. Porém, para os passos 4 e 6, uma análise mais profunda se faz necessária. Optamos por focar no passo 4, por acreditarmos que esta tarefa é a que pode apresentar o maior desafio para a automatização do método como um todo.

A fim de auxiliar e agilizar a realização dos passos que compõe a abordagem de Castro et al. (2011), seria interessante o uso de algum procedimento automatizado. Em seu trabalho, Castro chegou a analisar uma ferramenta para anotação semântica, o Palavras (http://visl.sdu.dk/visl/pt/). Apesar de identificar classes gramaticais e funções sintáticas corretamente, ocorreram falhas quando se atingiu o nível semântico. Neste trabalho é feita uma análise mais abrangente sobre esta classe de ferramentas.

Inicialmente, analisamos ferramentas de processamento de linguagem natural que realizam anotação sintática de partes do discurso (*Part-of-Speech Tagging*), ou seja, o processo de classificar palavras morfologicamente de acordo com as classes gramaticais [Bird *et. al*., 2009]. Em seguida, buscamos abordagens que realizam algum tipo de anotação semântica (*semantic tagging*) para classificação das palavras, considerando informações de contexto.

3.1. Abordagens de *POS Tagging*

O estudo realizado consistiu em aplicar algoritmos de *POS tagging* a uma lista de sentenças simples tratadas. Foram testados diferentes *frameworks* como o NLTk (http://nltk.org), o Apache OpenNLP (http://opennlp.apache.org/) e o Stanford Parser/Tagger (http://nlp.stanford.edu/). O primeiro possibilitou a aplicação dos algoritmos de *POS-tagging* N-Gram (considerando suas variações Unigran, Bigram e Trigram) e Brill Tagger [Brill, 1992]. O segundo e o terceiro implementam o Modelo de Entropia Máxima (*Maximum Entropy Model* ou MaxEnt Model) [Ratnaparkhi, 1996].

3.1.1. Análise Prática dos Algoritmos de POS Tagging

Os três algoritmos (N-Gram, Brill e MaxEnt) foram aplicados a um mesmo conjunto de 56 sentenças já processadas e resultantes do estudo de caso de Castro et al. (2011) após decomposição em sentenças simples, divisão em orações, depassivização, separação em sentenças nucleares, e criação e esclarecimento de lista de dúvidas. O objetivo era analisar as anotações geradas pelos algoritmos para os substantivos identificados por Castro como os signos importantes do universo a ser modelado.

Os três algoritmos foram capazes de anotar corretamente a maior parte dos termos das sentenças, identificando suas classes gramaticais com precisão similar à indicada pelos seus criadores. Entretanto, o resultado dos *taggers* se mostrou tão limitado quanto o da ferramenta Palavras, uma vez que os tipos semânticos não puderam ser especificados. Os corpora não contêm, em seu conjunto de *tags* possíveis, indicações de tipos semânticos. A Figura 1 exemplifica a aplicação do Stanford Tagger, onde observa-se que, que mesmo capaz de identificar o substantivo "professor", o *tagger* que não faz qualquer indicação sobre este ser um *Human* e mais especificamente um *Rank*. Sem um corpus que indique tais classificações, a análise sobre a capacidade dos *taggers* em reconhecer o contexto semântico não se fez possível.

Sentença Original	***"O Professor responsável pela disciplina defere ou indefere os requerimentos de inscrição em disciplina isolada."***
Stanford Tagger	`[('O', [art]) ('professor', [n]) ('responsável', [adj]) ('pela', [v-fin]) ('disciplina', [n]) ('defere', [v-fin]) ('ou', [conj-c]) ('indefere', [v-fin]) ('os', [art]) ('requerimentos', [n]) ('de', [prp]) ('inscrição', [n]) ('em', [prp]) ('disciplina', [n]) ('isolada', [v-pcp]) ('.', [punc])]`

Figura 1 - Exemplo de *taggeamento* realizado pelo Stanford Tagger

3.2. Abordagens de *Semantic Tagging*

Uma vez observado que os *taggers* atualmente utilizados pela comunidade de NLP não eram capazes de identificar os tipos semânticos dos signos no texto, optou-se por estender a pesquisa para abordagens que considerassem a anotação semântica (*semantic tagging*). O que foi percebido com a pesquisa é que o termo "*semantic tagging"* tem sido utilizado para denotar diferentes técnicas em áreas de atuação distintas, desde as que de fato abordam o processamento de linguagem natural até aquelas que tangem outras tarefas como a de criação automática de taxonomias ou apoio a criação de *folksonomias*. Ainda que não tenham sido encontrados trabalhos que permitissem a automatização da identificação dos tipos semânticos propostos por Dixon (2005) a partir dos signos de um texto, alguns trabalhos podem indicar caminhos interessantes a serem seguidos na busca pelo desenvolvimento (ou adaptação) de um *tagger* semântico.

Alguns trabalhos como os de Mahar e Memon (2010) e Miller *et. al*. (1993), propõe a utilização de bases de dados léxicas como o WordNet para restringir a semântica de termos de acordo com os outros termos presentes nas sentenças, o que poderia auxiliar no desenvolvimento de um *tagger* semântico. Gelbukh e Kolesnikova (2012) também fazem uso do WordNet ao introduzirem o problema de se reconhecer, automaticamente, o sentido das palavras participantes de "*collocations"* (colocações), situação aonde ao se combinar duas palavras uma delas tem seu valor semântico alterado. Buitelaar (1997) propõe o léxico CoreLex para *tagging* semântico, capaz de classificar informações em um nível semântico utilizando conceitos compatíveis com os construtos da OntoUML utilizados na abordagem de Castro et al. (2011).

4. Considerações e Trabalhos Futuros

Este trabalho analisou técnicas utilizadas atualmente pela comunidade de processamento de linguagem natural observando o quanto elas podem apoiar um método de modelagem conceitual com foco semântico, especificamente o proposto por Castro et al. (2011). O objetivo deste método é elaborar uma ontologia como modelo

conceitual de domínio semanticamente expressivo. O método é baseado em linguística e gera um modelo conceitual bem fundamentado em OntoUML a partir de textos em linguagem natural. A aplicação de tipos semânticos possibilita o aumento no poder de expressão e diminui as chances de erros decorrentes do processo de modelagem.

Foram aplicadas técnicas de anotação gramatical através dos *frameworks* NLTk, OpenNLP e Stanford Tagger. Foi observado resultado similar ao descrito por Castro et al. (2011) quando a ferramenta Palavras foi analisada. Optou-se então por buscar soluções alternativas com foco em *tagging* semântico e alguns trabalhos relacionados ao tema foram revistos, com suas possíveis colaborações explicitadas.

Não foi encontrada uma solução para *tagging* semântico que possa ser diretamente aplicada ao problema aqui abordado, entretanto foi possível detectar tópicos que combinados e estendidos poderiam auxiliar na automatização do método de Castro et al. (2011). Outras abordagens como Desambiguação de Sentidos de Palavras (*Word Sense Desabiguition*) podem também ser de grande ajuda e deverão ser objetos de estudo em trabalhos futuros. É importante ressaltar que apesar das técnicas aqui relatadas não serem suficientes de maneira isolada para a automatização da abordagem, elas podem diminuir o número de falhas no processo de modelagem conceitual, uma vez que a análise de textos em linguagem natural pode ser uma tarefa complexa até mesmo quando os indivíduos trabalham com suas línguas maternas.

Agradecimentos

O primeiro autor agradece ao Programa CAPES/REUNI pela bolsa de estudos recebida.

Referências

Bird, S., Klein, E., Loper, E. (2009) “Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit”. O’Reilly Media.

Brill, E. (1992) “A simple rule-based part of speech tagger”. In Proceedings of the third conference on Applied natural language processing (ANLC '92). Association for Computational Linguistics, Stroudsburg, PA, USA, 152-155.

Buitelaar, P. (1997) “A Lexicon for Underspecified Semantic Tagging”. Proceedings of ANLP 97, SIGLEX Workshop.

Castro, L., Baião, F. A., Guizzardi, G. (2010) “A Linguistic Approach to Conceptual Modeling with Semantic Types and OntoUML”. EDOCW 2010: 215-224.

Castro, L., Baião, F. A., Guizzardi, G. (2011) “A Semantic Oriented Method for Conceptual Data Modeling in OntoUML Based on Linguistic Concepts”. ER 2011: 486-494.

Dixon, R. (2005) “A Semantic Approach to English Grammar”. Oxford University Press.

Gelbukh, A., e Kolesnikova, O. (2012) “Supervised Learning Algorithms Evaluation on Recognizing Semantic Types of Spanish verb-noun Collocations”. Computación y Sistemas.

Guizzardi, G. (2005) “Ontological Foundations for Structural Conceptual Models”. Ph.D. dissertation, University of Twente, Enschede, The Netherlands.

Guizzardi, G., das Graças, A. P., Guizzardi, R. S. S. (2011) “Design Patterns and Inductive Modeling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML”. CAiSE Workshops 2011: 402-413.

Mahar, J. A., e Memon, G. Q. (2010) “Sindhi Part of Speech Tagging System using Wordnet”. International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 2010

Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T. (1993) “A Semantic concordance”. Proceedings of the workshop on Human Language Technology.

Ratnaparkhi, A. (1996) “A Maximum Entropy Model for Part-Of-Speech Tagging”. Department of Computer and Information Science – University of Pennsylvania.

Uma Ontologia das Classificações da Despesa do Orçamento Federal[1]

Luís Sérgio de O. Araújo[1], Daniel Aguiar da Silva[1], Mauro T. Santos[1], Fernando W. Cruz[2], Matheus S. Fonseca[2], Guilherme de L. Bernardes[2]

[1]Secretaria de Orçamento Federal – SOF/MP
SEPN 516, bloco D, lote 8 CEP: 70.770-524 – Brasília – DF - Brasil

[2]Laboratórios de Engenharia e Inovação – FGA – UnB (Universidade de Brasília)
Caixa Postal 15.064 – 71.501-970 – Brasília – DF – Brasil

{luis.araujo,daniel.aguiar,mauro.santos}@planejamento.gov.br, fw-cruz@unb.br, matheus.souza21@live.com, guilhermedelima@hotmail.com

Abstract. *This paper presents in OWL language the Federal Budget Expenditures Classification Ontology which aims at enabling developers and public finance experts a complete, unrestricted and automatic access to the brazilian federal budget data.*

Resumo. *Este artigo apresenta, em OWL, a Ontologia da Classificação da Despesa do Orçamento Federal, que tem como propósito possibilitar à comunidade de desenvolvedores e técnicos em finanças públicas, o acesso completo, irrestrito e automático aos dados do orçamento federal brasileiro.*

1. Introdução

O orçamento público nasceu com a necessidade de controlar a arrecadação e os gastos dos governos, pelo parlamento. Decorre daí um princípio elementar de finanças públicas, para o qual nenhuma despesa pública pode ser realizada sem autorização legislativa. Inspirados na Carta Magna Inglesa de 1215, os orçamentos modernos passaram a ter previsão de receita e despesa anual obrigatória em Lei.

No Brasil, a LOA – Lei Orçamentária Anual é, ao mesmo tempo, instrumento de gestão e de transparência, dando previsão das ações planejadas pelo governo para um exercício financeiro, que, no Brasil, corresponde ao período de 1º de janeiro a 31 de dezembro.

Não obstante as nobres origens históricas, na prática a complexidade e extensão do orçamento – 2.645 páginas, no caso do orçamento 2012 (Brasil, 2012) – dificultam sua efetiva transparência e acompanhamento. Enquanto as áreas técnicas do governo dispõem de sistemas de informações que lhes permitem produzir relatórios com informa-

[1]Este trabalho foi produzido no âmbito do Termo de Cooperação MP-UnB sob a coordenação, no MP, do secretário-adjunto da SOF Eliomar Wesley Rios e na UnB, do professor Rafael Timóteo de Souza Junior. O projeto foi desenvolvido na Coordenação-Geral de Tecnologia e Informações da SOF-MP sob a direção de Carlos Eduardo Lacerda Veiga, pela SOF e Daniel Alves da Silva, pela UnB.

ções agregadas e seletivas, de forma a subsidiar as decisões dos gestores públicos, o cidadão comum, por sua vez, não dispõe de recursos equivalentes.

A fim de aumentar a transparência e acesso ao orçamento, alguns esforços vêm sendo despendidos pelo governo federal (Brasil, 2012a), sem contudo eliminar a assimetria de informações criada pela falta de instrumentos que permitam ao cidadão um nível de acesso mais amplo e livre.

Neste sentido, a Secretaria de Orçamento Federal do Ministério do Planejamento, Orçamento e Gestão – SOF/MP, motivada pela nova Lei de Acesso à Informação (Brasil, 2011), que, entre outros aspectos, em seu art. 8º, §3º preconiza a abertura de dados governamentais de forma estruturada e legível por máquina, iniciou esforços para estabelecer um mecanismo que permita à sociedade civil organizada o efetivo acesso às infor mações orçamentárias do Governo Federal.

A estratégia adotada foi criar uma ontologia da classificação da despesa do orçamento federal, contemplando as categorias e conceitos sedimentados no MTO (Brasil, 2012b) os quais resultam de décadas de evoluções da doutrina do orçamento público e de debates entre especialistas, parlamentares, gestores e representantes da sociedade. Utilizamos, aqui, o termo ontologia com o sentido de especificação – mais utilizada na ciência da computação – em contraste com o sentido de sistema conceitual – normalmente usado na área de sistemas de informação (Hepp, 2007). O propósito imediato do trabalho neste caso é, primordialmente operacional, qual seja, viabilizar a publicidade dos dados em formato computacionalmente tratável.

O trabalho é dirigido à comunidade de desenvolvedores, que poderá utilizá-lo para criar produtos voltados aos técnicos em finanças públicas e aos cidadãos comuns, destinatários finais desses esforços, considerando suas idiossincrasias. Um tratamento deste domínio referenciando a visão do cidadão comum é, sem dúvida, um *desideratum* desafiador. Acreditamos que este trabalho colabora com este objetivo, que deverá ser perseguido em desenvolvimentos futuros.

Este artigo está assim estruturado: a Seção 2 traz uma breve explanação sobre a estrutura de organização do orçamento federal, para melhor situar o leitor não versado no tema; a seção 3 descreve a ontologia, sua forma de implementação e tecnologias utilizadas; a seção 4 traz as considerações finais e conclusões.

2. Estrutura do Orçamento Público Federal

A base para a compreensão do orçamento público é o sistema de classificação. É por meio desse sistema, que o orçamento é organizado, ou seja, segmentado com base em critérios. Essa estrutura permite que os técnicos e gestores públicos consigam estratificar os dados e estabelecer as relações entre os valores financeiros do orçamento e os fenômenos da administração pública associados (e.g. gasto em quê, para quê, sob a responsabilidade de quem).

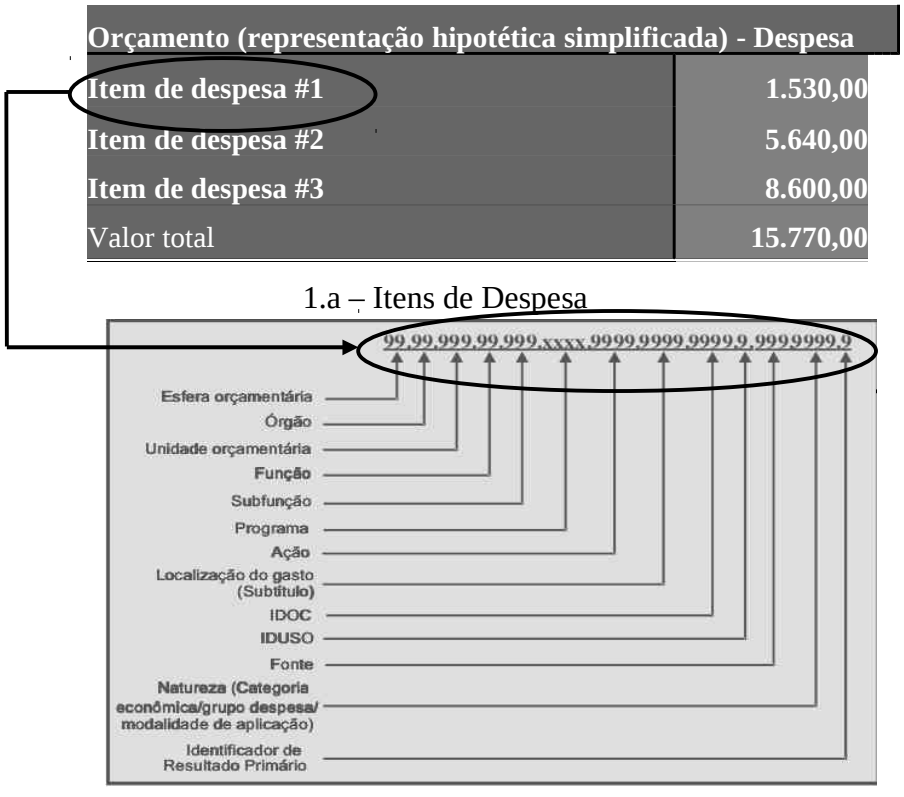
Jesse Burkhead (Burkhead, 1956), propôs quatro objetivos essenciais para um sistema de classificação: 1) facilitar a formulação de programas; 2) contribuir para a execução

do orçamento, 3) fixar responsabilidades e 4) possibilitar a análise dos efeitos econômicos das atividades governamentais (Teixeira Machado, 1967, pág 64).

O orçamento público apresenta uma representação detalhada da receita e da despesa de um ente público (União, estado ou município) para o período de um ano. Existe um sistema de classificação específico para a receita e outro para a despesa. Por limitação de espaço, este trabalho focaliza apenas a despesa, por ser o aspecto normalmente mais discutido do orçamento.

A estrutura orçamentária é composta basicamente de uma lista de itens de despesa com seus respectivos valores. Cada item é representado por um código de 37 dígitos, estruturado em 13 campos, que o vinculam aos critérios de classificação despesa.

A Figura 1 apresenta uma representação simplificada de um orçamento fictício composto por três itens de despesa. No caso do orçamento federal, este número chega a dezenas de milhares de itens. A Figura 1.b apresenta o esquema da classificação da despesa, com os critérios referentes a cada um dos 13 campos que compõem o seu código. A título de exemplo, o campo "Órgão" classifica o item em relação ao órgão responsável. O significado de cada critério de classificação e os códigos correspondentes podem ser encontrados no Manual Técnico de Orçamento 2012 – MTO 2012 (Brasil, 2012b).



1.a – Itens de Despesa

1.b – Critérios de classificação de itens de despesa orçamentária (Brasil, 2009)

Figura 1 – Representação simplificado hipotética de um orçamento

Essa estrutura permite que sejam respondidas diversas perguntas, a exemplo de valores gastos por um determinado órgão ou nos vários programas e ações de governo.

3. Ontologia das Classificações da Despesa do Orçamento Federal 2012

A ontologia descrita é composta pelos itens de despesa e os seus classificadores orçamentários (e.g. Função, SubFunção, Programa etc). Na Figura 2, os conceitos identificados são descritos como classes OWL, identificadas por retângulos na cor cinza, e as relações entre elementos de classes por retângulos pretos. O *range* das relações é representado por uma letra 'r', enquanto o *domain* é representado pela letra 'd'. Linhas tracejadas representam relações de subclasse. O prefixo siop[2] foi assumido para as propriedades de objeto (*object property*), enquanto as propriedades de tipos de dados (*data type property*) são representadas por retângulos brancos contendo seu tipo. Subentende-se aqui que os elementos da classe Item de Despesa são definidos por terem uma relação unívoca (1:1) com um elemento de cada uma das classes referentes aos diferentes tipos de classificadores vigentes no orçamento. Apesar de não constar, estão implícitas: (i) a classe Classificadores, que reúne todos os classificadores orçamentários apresentados na Figura 2, e (ii) a classe Orçamento Anual, que contempla os itens de despesa, dentre outros elementos.

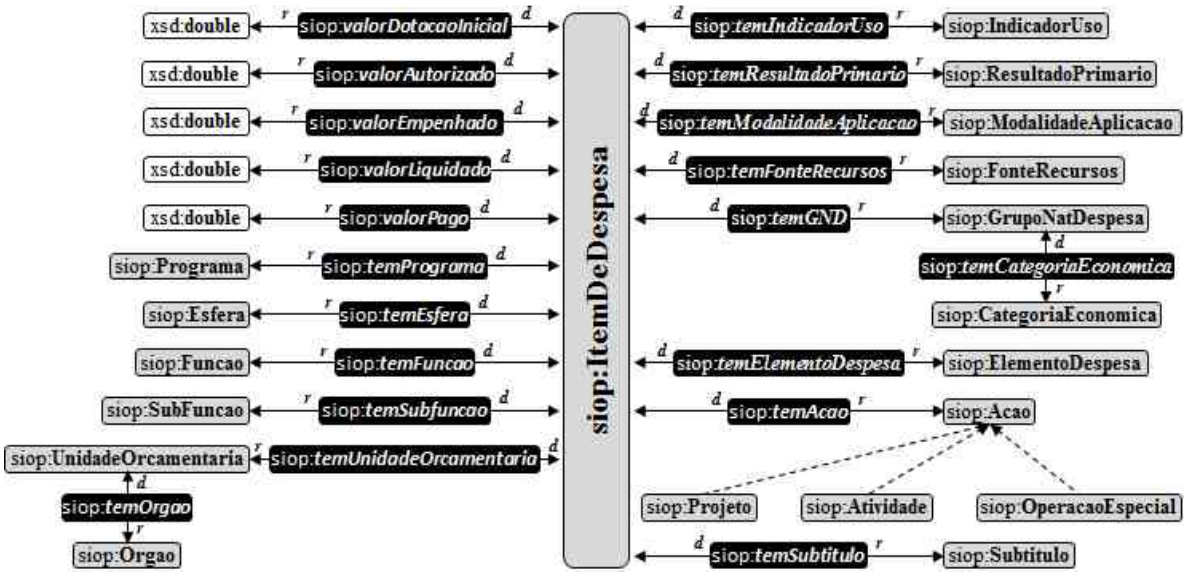


Figura 2 – Ontologia das Classificações da Despesa do Orçamento Federal

Como prova de conceito, essas classes foram utilizadas para converter as informações constantes nos bancos de dados relacionais (Figura 3.a) em triplas RDF (Figura 3.b), tomando como base os conceitos definidos na ontologia (Figura 3.c). A Figura 3, no entanto, não contempla a totalidade das classes consideradas. O processo de triplificação envolveu várias fases e foi realizado pela adaptação da ferramenta Triplify, versão 0.8 (triplify.org), utilizada na conversão para o formato *N-Triples*, e pelo tratamento dos ca-

[2]SIOP – Sistema Integrado de Planejamento e Orçamento, do Ministério do Planejamento.

racteres especiais (vogais acentuadas, cedilha, etc). Em paralelo a esse processo, foi feita uma investigação para identificar um *endpoint* para consultas SPARQL sobre os dados convertidos. Dentre os ambientes testados, optou-se pelo Fuseki, versão 0.2.2 (http://jena.apache.org/documentation/serving_data/index.html). Além disso, foi desenvolvida uma aplicação com consultas federadas (www.siop.planejamento.gov.br) para demonstrar a possibilidade de cruzamentos de dados orçamentários com informações constantes em bases como, por exemplo, Google Maps e Dbpedia, sendo este um aspecto especialmente promissor para as potenciais aplicações desta ontologia.

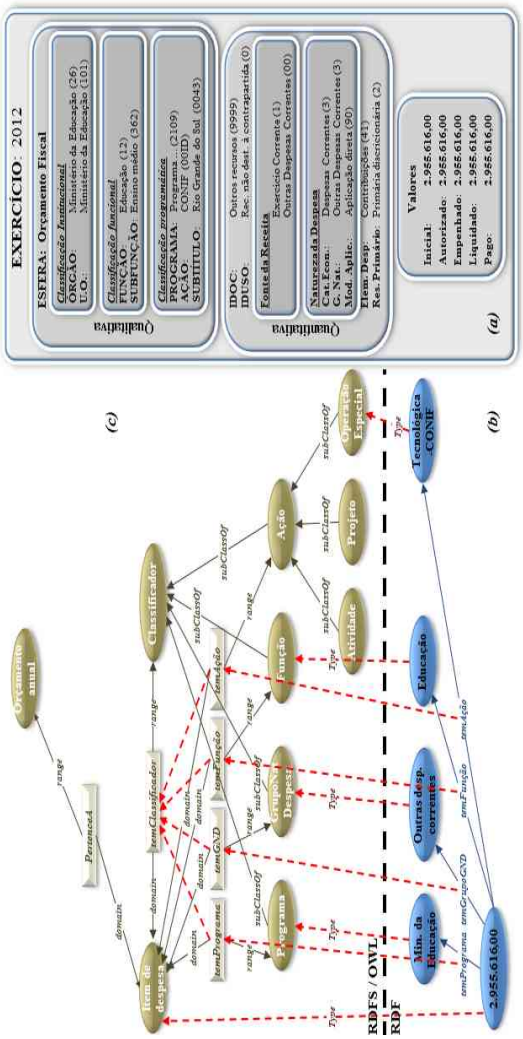


Figura 3 – Processo de preenchimento da ontologia

4. Considerações Finais

A ontologia populada com os dados da Lei Orçamentária Federal 2012 gerou um arquivo rdf com 824.591 triplas, publicado com a especificação owl e um manual de referência no portal www.siop.planejamento.gov.br, em 30/08/2012, para o acesso irrestrito às

informações da despesa prevista no orçamento, segundo os critérios do sistema de classificação. Sua utilização pode ser automatizada utilizando a linguagem SPARQL ou uma biblioteca adequada como, por exemplo, JENA. Desenvolvedores podem fazer uso imediato da ontologia e dos dados associados. Trata-se portanto de um produto pronto para uso, porém não acabado. A ontologia poderá ser acrescida com mais informações, tais como a descrição dos programas de governo, o acompanhamento da execução, relatórios de auditoria, legislação, definições, exercícios anteriores, entre outros.

A literatura acadêmica pouco tem abordado o tema ontologia-orçamento público. Esperamos que a divulgação desta ontologia e dos dados correspondentes possa fomentar o interesse e o diálogo na comunidade.

Entre as possibilidades que se apresentam diante deste passo está o desenvolvimento, pela comunidade de desenvolvedores, de produtos diversos que poderão fazer o cruzamento dos dados do orçamento federal com os de estados e municípios, e outros de naturezas diversas, a exemplo da correlação entre os recursos destinados à saúde de uma determinada região e a evolução dos respectivos indicadores sociais.

Essencialmente, este trabalho vai ao encontro das iniciativas do governo brasileiro no campo da transparência e do acesso público às informações governamentais de interesse geral. Espera-se que a iniciativa produza um debate construtivo na comunidade de ciência da computação e de finanças públicas sobre as possibilidades e a contribuição das novas tecnologias para o aperfeiçoamento da democracia no país.

Referências

Brasil (2011) Lei nº 12.527, de 18 de novembro de 2011. http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2011/Lei/L12527.htm

Brasil (2012) Lei nº 12.595, de 19 de janeiro de 2012. http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12595.htm

Brasil (2012a) “Orçamento Federal ao Alcance de Todos. Projeto de Lei Orçamentária Anual – PLOA”, Brasília: Ministério do Planejamento, Orçamento e Gestão.

Brasil (2012b) “Manual Técnico Orçamentário”, Versão 2012, Brasília: Ministério do Planejamento, Orçamento e Gestão – Secretaria de Orçamento Federal.

Brasil (2009) “Manual Técnico Orçamentário”, Versão 2009, Brasília: Ministério do Planejamento, Orçamento e Gestão – Secretaria de Orçamento Federal.

Burkhead, Jesse. Government Budgeting. London: John Wiley and Sons, Inc., 1956.

Hepp, Martin at al; Ontology Management: Semantic Web, Semantic Web Services, and Business Aplications (Semantic Web and Beyond), Springer, 2007, Kindle edition.

Teixeira Machado, J. Jr. Classificação da Contas Públicas. Rio de Janeiro: Fundação Getúlio Vargas, 1967.

Modelando Ontologias a partir de Diretrizes Clínicas: Diagnóstico e Tratamento da Cefaléia

Eduardo J. Zanatta[1], Fabrício H. Rodrigues[2], Silvio C. Cazella[1], Cecília D. Flores[1], Marta R. Bez[2]

[1]Universidade Federal de Ciências da Saúde de Porto Alegre - UFCSPA

[2]Univerisdade FEEVALE.

eduzanatta@gmail.com, fabriciohr@feevale.br, silvioc@ufcspa.edu.br, dflores@ufcspa.edu.br, martabez@feevale.br

Abstract. *This paper presents a methodology used to develop ontologies based on clinical guidelines, which consists in adapting the method 101 for adapting it to obtain ontologies from medical knowledge structured in a textual form. To do so, it is applied some adjustments to the steps presented in the original methodology, based on some guiding principles for the process. In order to test the methodology adapted, an ontology is built for the domain of diagnosis and treatment of headache.*

Resumo. *O presente artigo apresenta uma metodologia utilizada no desenvolvimento de ontologias tendo como base diretrizes clínicas, que consiste na adaptação da metodologia 101 para adequá-la à obtenção de ontologias a partir de conhecimento médico estruturado em formato textual. Para tanto, são aplicadas algumas adaptações aos passos apresentadas na metodologia original, tendo por base alguns princípios norteadores para o processo. Para experimentar a metodologia adaptada, foi construída uma ontologia para o domínio de diagnóstico e tratamento de cefaleia..*

1. Introdução

O presente artigo consiste em apresentar a metodologia utilizada no desenvolvimento de ontologias tendo como base de informação o domínio de medicina de família e comunidade, mais especificamente uma diretriz clínica sobre cefaleia (Wagner H. et al, 2012).

Diretrizes clínicas consistem em normativas estabelecidas por um grupo de especialistas colimados por notório saber de uma área, que se propõem a reunir e emitir um consenso sobre as melhores práticas do estado atual do conhecimento de alguma questão clínica. Essa revisão é geralmente embasada em literatura técnica atualizada, associada à opinião e experiência desse grupo de peritos. Essas diretrizes clínicas são então divulgadas geralmente por associações de especialistas, com periódicas revisões de atualização.

O interesse neste processo está em buscar um meio de facilitar a geração de redes bayesianas, baseadas em diretrizes clínicas, a partir das ontologias criadas. Sendo assim, o trabalho aqui apresentado compreende a formalização em termos computacionais de conhecimento médico expresso em linguagem natural, de forma a

propiciar os benefícios que a utilização desta tecnologia poderá trazer ao projeto de simulações médicas baseadas em conhecimento incerto - SimDeCS (Flores et al, 2012).

O artigo está estruturado de forma a apresentar o desenvolvimento da ontologia através de uma adaptação da metodologia 101, aspectos considerados importantes, seguido das considerações finais.

O restante deste trabalho está estruturado da seguinte forma: na seção 2 traz o relato do desenvolvimento da ontologia, apresentando a metodologia empregada e na seção 3 são apresentadas algumas considerações finais..

2. Desenvolvimento da Ontologia

Representações ontológicas do conhecimento médico despertam interesse de diversos grupos de pesquisa. (Preços e Spackman, 2000), (NCBO, 2011), (Humphreys e Lindberg, 1993). Diversas ontologias estão disponíveis em repositórios na web, porém, comumente desenvolvidas para uso de alguma aplicação específica, o que pode afetar o desempenho desta quando usada por outra aplicação (Baader et al, 2003). Optou-se assim pelo desenvolvimento de uma nova ontologia que expresse o conhecimento contido na Diretriz Clínica escolhida, consenso pela Sociedade de Medicina de Família e Comunidade brasileira.

Distintas metodologias para construção de ontologias, que focam sua construção desde o início, são encontradas na literatura, tais como Método Cyc (Reed e Lenat 2002), Método Sensus (Swartout et al, 1996), Methontology (Fernandez, Gomez Pereze e Juristo, 1997). Outro guia para desenvolvimento de ontologias é o método 101, (Noy e McGuinness, 2001). Este guia foi concebido utilizando o software Protégé-2000.

Devido à falta de uma metodologia consensual para a criação de ontologias, muitos pesquisadores seguem seus próprios critérios de desenvolvimento. Entretanto, parece razoável considerar que, em metodologias cuja capacidade geral de guiar o processo de construção de ontologias já tenha sido atestada, a inserção de critérios adicionais e adaptação de etapas pode prover maior adequação à situações determinadas. Por outro lado, em domínios menores e cujas fontes de conhecimento são parcialmente estruturadas – como no caso de diretrizes clínicas – o impacto negativo desse tipo de abordagem pode ser mitigado, prevalecendo seus aspectos positivos.

Sendo assim, e considerando os objetivos do trabalho, para construção de um modelo de ontologia que descreva o conhecimento contido na diretriz clínica, uma adaptação da metodologia 101 para construção de ontologias (NOY; MCGUINNESS, 2011) foi utilizada. Essa foi eleita como modelo a ser adaptado devido, principalmente, à sua simplicidade, estando acessível de imediato pelos membros da equipe e implicando em facilidade de aplicação. Além disso, por seu carácter menos estruturado e, consequentemente, mais flexível que as outras metodologias estudadas, a modelagem de uma diretriz clínica na forma de uma ontologia torna-se mais natural, não suscitando conflito com etapas da metodologia.

O método 101 é fundamentalmente baseado em quatro etapas: i) definição das classes; ii) organização das classes em taxonomia; iii) definição de relações, atributos e os valores que cada um poderá receber; iv) criação de instâncias pela inserção de valores correspondentes a cada atributo. A metodologia aponta, de forma mais específica, 7 passos imprescindíveis para a construção de ontologias. Na aplicação

desses passos foram observados 3 princípios para adequação ao uso sobre diretrizes clínicas:

(a) determina que os termos utilizados na ontologia tenham origem exclusivamente no texto da diretriz clínica, a exceção de conhecimento tácito inerente ao domínio médico. (b) que todo o conhecimento presente no texto da diretriz deve ser modelado na ontologia, a fim de que se tenha uma representação fiel do domínio descrito em seu texto.

(c) indica a utilização de algumas superclasses que representem estereótipos médicos recorrentes (i.e Diagnóstico, Indício, Sintoma, Sinal, Tratamento, Combate, Profilaxia e Contexto).

A aplicação à diretriz clínica dos 7 passos referidos no modelo 101 ocorreu como segue:

I) Determinar o domínio e escopo da ontologia: Definição dos princípios para a construção da ontologia. Também foram formuladas as chamadas "questões de competência", que representam perguntas às quais a ontologia deve ser capaz de responder. Essas questões, além de guiar o projeto, servem para a validação da ontologia, permitindo verificar se ela atende aos objetivos para que foi construída. Entre elas estão perguntas tais como "Quais são os tipos de tratamento recomendados para enxaqueca?", "Que sintomas caracterizam cefaleia em salvas?" e "Dados esses sintomas, que exame complementar deve ser solicitado?". A participação de um especialista do domínio médico foi necessária dada a especificidade e complexidade do domínio em questão.

II) Considerar o reuso de ontologias existentes: verificação da existência de termos e/ou porções do conhecimento em outras ontologias que possam ser reutilizadas na ontologia em desenvolvimento. Obedecendo ao princípio (a), não foram buscadas ontologias referentes a cefaleia, sendo todos os termos modelados a partir da interpretação da diretriz clínica e da interação com o especialista de domínio.

III) Enumerar os termos importantes da ontologia: relação dos termos que representam conceitos do domínio, bem como os úteis para descrever tais conceitos. Dado o princípio (a), o processo de enumeração de termos ficou constrito ao vocabulário utilizado no texto da diretriz clínica, com acréscimo apenas do conhecimento médico implícito identificado pelo especialista. Nesse processo, diferentemente do recomendado na metodologia 101, foram classificados os termos, conforme o tipo de elemento da ontologia a que correspondiam. Essa estratégia foi adotada tendo em vista a estrutura semântica do vocabulário imposta pelo texto da diretriz, que evidencia o papel de cada termo dentro do domínio.

IV) Definir a hierarquia de classes: organização dos termos listado definindo-se classes e subclasses. Por questões de engenharia da ontologia, decidiu-se por criar algumas classes-raiz para agrupar conceitos da forma como tradicionalmente são interpretados no domínio médico. Assim, foram criadas as seguintes classes: Diagnóstico, Indício, Sintoma, Sinal, Sinal/Sintoma de alerta, Exame complementar, Tratamento, Contexto, Perfil, Histórico, Fator externo. A partir destas, os conceitos identificados foram classificados, adicionando subclasses necessárias para representar a hierarquia inferida.

V) Definir propriedades e relações: os termos não utilizados na etapa anterior foram, em sua maioria, definidos como propriedades e relações das classes, aprofundando a representação do domínio e permitindo responder às questões de competência mais satisfatoriamente. Com o objetivo de criar um dicionário de sinônimos e disponibiliza-lo para uso em outras ontologias, termos similares foram agrupados. Cabe ressaltar a importância do especialista de domínio nessa fase.

VI) Definir as características das propriedades e relações: criação de definições relativas a cardinalidade, tipo, domínio e contradomínio de cada propriedade e relação. Com isso foram formadas diversas expressões representando as relações entre classes e a sua caracterização por propriedade ("náusea evidencia enxaqueca", que traz a relação "evidencia" entre os conceitos "náusea" e "enxaqueca").

VII) Criar instâncias: representação dos objetos pelo preenchimento das propriedades e relações das classes. Identificou-se na diretriz termos ou expressões que representassem exemplos dos conceitos modelados. Esse processo teve por base a granularidade evidenciada pela diretriz (os conceitos que não poderiam ser divididos em subclasses e dos quais não faria sentido ou não seria útil a criação de instâncias) e, em geral, identificou como instâncias expressões formadas por conceitos, propriedade e valores de propriedades ("dor de intensidade moderada").

Tratando-se de um processo iterativo, a identificação dos termos da ontologia deste trabalho teve duas iterações. Na primeira, descrito acima, foram identificados os termos expressos diretamente em algum termo do texto (e.g. dor, cefaleia, secreção nasal). Na segunda, adotou-se uma visão mais abrangente, objetivando identificar termos inferíeis do contexto expresso por fragmentos maiores do texto (e.g. o fragmento "O diagnóstico de enxaqueca e cefaleia tensional em idosos é, muitas vezes, um desafio, tendo em vista que o início dos sintomas depois dos 50 anos é infrequente e pode representar uma cefaleia de origem secundária como, por exemplo, massas expansivas intracranianas e acidente vascular cerebral. Existe a necessidade de uma atenção maior nesta faixa etária, já que as causas secundárias de cefaleia são mais prováveis" permitiu identificar relações como "idade maior que 50 anos evidencia cefaleia secundária" e "acidente vascular cerebral causa cefaleia secundária").

A Figura 1 trata de um fragmento da ontologia, apresentando apenas a hierarquia de suas classes principais. Pode ser visualizada a classe "Thing", representando a superclasse de todas as coisas. Um nível abaixo encontram-se as 4 classes definidas no quarto passo da metodologia ("Diagnostico", "Evidencia", "Contexto" e "Tratamento"). O nível seguinte mostra as subdivisões das classes "Evidencia" e "Contexto" ("Sintoma", "Sinal", "SinalSintomaAlerta", "ExameComplementar", "Perfil", "Historico" e "FatorExterno"). Também podem ser visualizadas as classes "Cefaleia" e suas subdivisões, "CefaleiaPrimaria" e "CefaleiaSecundaria", bem como alguns tipos de cefaleia primaria (i.e. "Enxaqueca", "CefaleiaEmSalvas" e "CefaleiaTensional"), que representam o cerne do domínio da ontologia. Há ainda uma classe para representar conhecimento de mais alto nível – "TipoDor" – que, formando uma partição de valor, especifica, em uma possível instância da classe "Dor", o tipo da dor representada pela instância – seja "Pulsatil" ou "Pressao".

Figura 1. Fragmento da Ontologia de Cefaleia

3. Considerações Finais

A construção de ontologias a partir de diretrizes clínicas parece ser uma boa abordagem. Apesar da relativa complexidade do modelo criado em comparação ao tamanho da diretriz que serviu como base (i.e. mais de 180 classes extraídas de uma diretriz de 13 páginas) a forma esquemática como o conhecimento é organizado na ontologia torna sua consulta mais objetiva tanto com uso de métodos físicos tradicionais (i.e. análise de diagramas representativos da ontologia) quanto daqueles intermediados por software. Além disso, nesse último caso, torna-se possível a aplicação de recursos para potencialização do conhecimento armazenado (e.g. interface de comunicação entre sistemas, sistemas de apoio a diagnóstico), os quais não estariam completamente

disponíveis para aplicação sobre a informação em linguagem natural. Dessa forma, a conversão de diretrizes clínicas em ontologias torna o conhecimento mais disponível, com uso mais efetivo e, dada sua formalização, mais facilmente padronizável – e sendo essas características-chave tanto da natureza quanto dos objetivos de uma diretriz clínica, fortalece-se o argumento em favor dessa prática.

Alinhado a isso, como trabalho futuro, tem-se uma validação formal e de domínio mais refinada, com a comparação objetiva entre a ontologia e a diretriz original a fim de determinar em que medida elas são representações correspondentes. Ainda, esse trabalho está conectado a outro cujo objetivo é a extração, a partir de ontologias médicas, de redes bayesianas para diagnóstico, estando ambos ligados ao projeto do SimDeCS [12].

4. Referências

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (2003) " The Description Logic Handbook".

Fernandez, M. Gomez-Perez, A. Juristo, H. (1997) " Methontology: from ontological art twards ontological engineering".

Flores, C., D. Bez, M.,R. Respicio, A. Fonseca, J., M. (2012) "Training Clinical Decision-Making through Simulation".

Humphreys, B., L. Lindberg, D., A., B. (1993) "The UMLS project: making the conceptual connection between users and the information they need". Bulletin of the Medical Library Association.

NCBO. BioPortal. (2011) "The National Center for Biomedical Ontology". http://bioportal.bioontology.org/, Agosto.

Noy, F., N. Guinness, D., L, (2001) "Ontology development 101: a guide to create your first ontology".

Swartout, B. Patil, R. Knight, K. e Russ, T. (1996) "Toward Distributed Use of Large-Scale Ontologies".

Price, C. Spackman, K. SNOMAD. (2000) BJHC&IM-British Journal of Healthcare Computing & Information Management.

Reed S., L. Lenat B., D. (2002) "Mapping Ontologies into Cyc".

Wagner, H., L. Pinto, M., E., B. Klafke, A. Ramos, A. Stein, A., T. Castro Filho, E., D. (2012) "Diagnóstico e tratamento das cefaleias em adultos na atenção primária à saúde". http://www.sbmfc.org.br/media/file/diretrizes/cefaleia.pdf, Janeiro.

Using Events from UFO-B in an Ontology Collaborative Construction Environment

Douglas Eduardo Rosa[1], **Joel Luis Carbonera**[1], **Gabriel M. Torres**[1], **Mara Abel**[1]

[1]Institute of Informatics – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{derosa,jlcarbonera,gmtorres,marabel}@inf.ufrgs.br

Abstract. *In previous works, it was introduced an approach to support collaborative construction and evolution of domain ontologies based on UFO-A. However, there are areas of knowledge, like in the petroleum geology, where we need to represent some natural/geological phenomena which aren't possible to be well represented by UFO-A constructs, by the fact that this concepts are treated as* events *that occur during time. Thus, this paper presents the results of the study to support events from UFO-B in an ontology collaborative construction environment, which initially was designed to maintain a collaboration record from a domain ontology based on UFO-A.*

1. Introduction

Recently, Torres' work [Torres et al. 2011, Torres 2012] has introduced an approach to building domain ontologies in a collaborative manner, in which the coherence in the negotiation of meaning during the collaboration is established with the use of a foundational ontology called UFO [Guizzardi 2005]. However, Torres' work just covers the fragment from UFO called UFO-A, which is an on ontology of *Endurants*, leaving aside the representation of events included in UFO-B (an ontology of *Perdurants*). UFO-A deals with objects that are wholly present whenever they are present, we say that the Endurants *are in time*, in the sense that, if in a circumstance c_1 an endurant e has a property P_1 and in a circumstance c_2 the property P_2 (probably incompatible with P_1), the object e is still the same endurant. On the other hand, Perdurants are individuals composed of temporal parts and, unlike Endurants, they *happen in time*, in the sense that they extend in time accumulating temporal parts [Guizzardi et al. 2008a]. For the record, examples of Endurants are a person, a car, a house, a rock; constitute example of Perdurants a conversation, a football game, a business process, a depositional process etc.

The Carbonera's work [Carbonera et al. 2011, Carbonera 2012] investigated the role played by foundational ontologies in the problem solving methods involving visual information, and proposed a cognitive model for visual interpretation of depositional processes, within the Sedimentary Stratigraphy domain. This work also developed a domain ontology in the area of Sedimentary Stratigraphy domain, in which were identified concepts that should be represented as Perdurants, i.e., it is required a representation of the temporal aspects in order to support reasoning.

So, there are knowledge domains in which their representation by a domain ontology requires the use of specific constructs that can represent temporal aspects, so this knowledge can be represented in a reliable way.

This paper presents the results of the study to support events from UFO-B in an ontology collaborative construction environment, which initially was designed to maintain a collaboration record from a domain ontology based on UFO-A. The choice of UFO-B, instead of UFO-C, comes from the fact that UFO-C represents intentional agents (that have *intentions* and *goals*), which is not our case. Our research group works with petroleum geology, so we do not want to represent intentional agents, we want to model geological and natural phenomena. This kind of phenomenon does not occur by the intention of an agent, but for natural reasons that do not have a specific goal. It is important to be clear that this work does not presents contributions in the creation of constructs for modeling events or temporal aspects in a domain ontology, but it is about the requirement that our modeling tool must have to support the collaborative construction of ontologies based on both concepts from UFO-A and UFO-B.

This paper is organized as follows: in section 2 are presented the main concepts envolving UFO. In section 3 we present the related works, which were the basis for the development of this work. In section 4 are presented the changes made in the architecture of the environment to support UFO-B events. In section 5, we conclude and anticipate some future work.

2. Foundational Ontology: UFO

The UFO (*Unified Foundational Ontology*) [Guizzardi 2005, Guizzardi et al. 2008a] is a Foundational Ontology which arose as a unification of concepts addressed by other foundational ontologies. UFO is divides into 3 parts - UFO-A, UFO-B and UFO-C - that structure notions of different scopes. The core of the UFO is the fragment called UFO-A, which is an ontology of *Endurants*, and is concerned with structural aspects, like objects, their types, their part/wholes, their intrinsic, relational properties and spaces of property values, distinction between different types and their allowable relationships etc. UFO-B is an ontology of *Perdurants* that deals with dynamic aspects, like events and their parts, relations between events, object participation in events, temporal properties of entities, time etc. UFO-C is built on top of UFO-A and UFO-B to systematize social aspects, like agents, intentional states, goals, actions, norms, social commitments among many others.

A complete and detailed description about UFO's fragments is outside the scope of this paper. For a more detailed description of UFO-A and UFO-B, including their logical meaning, refer to [Guizzardi 2005, Guizzardi et al. 2008a, Guizzardi et al. 2008b], in which the theoretical foundation of this work was based.

3. Collaborative Construction of Domain Ontologies

The work in [Torres et al. 2011] and [Torres 2012] presents an approach to build visual domain ontologies in a collaborative way using meta-data based on foundational ontologies. It makes use of the concepts from the foundational ontology UFO, seeking a well-structured construction of domain ontologies that could serve as a future reusable artifact. Furthermore, a goal of using foundational ontologies is to establish a theoretical basis to achieve consistency in the negotiations of meaning during the collaboration process. The collaborative aspect is motivated by the fact that knowledge domains are not static, they evolve as new elements become part of the domain or when elements become obsolete. These changes must be adapted to the domain model, updating the ontology by adding

or removing elements. As knowledge about a certain domain can belong to several geographically dispersed collaborators, a collaborative process to create and maintain an ontology helps to make explicit the changes that occur in this ontology, and also makes explicit the concepts involving the vocabulary used by different collaborators.

In order to support the creation of domain ontologies in a collaborative way, two meta-data ontologies are used: the $R.O.$ (Representation Ontology) and $C.O.$ (Collaboration Ontology). The $R.O.$ defines primitives to represent the domain ontology, while the $C.O.$ defines primitives to represent the collaboration events. The domain ontology components are defined as instances of the concepts of the $R.O.$, and the changes made in the domain ontology are defined as instances of the concepts of $C.O.$. The meta-ontology $R.O.$ extends some of the major components of ontologies (concept, property, relation, axiom) specializing them in the constructs proposed by foundational ontology UFO-A [Guizzardi 2005], providing thus an ontological foundation for the model. The meta-ontology $C.O.$ defines which events of collaboration can be performed in the domain ontology. The instances of $C.O.$ are the changes related to what is represented by the $R.O.$, which is the domain ontology. Figure 1 shows an example of the interaction in the ontology of meta-data and collaboration history generated due to changes in the domain ontology.

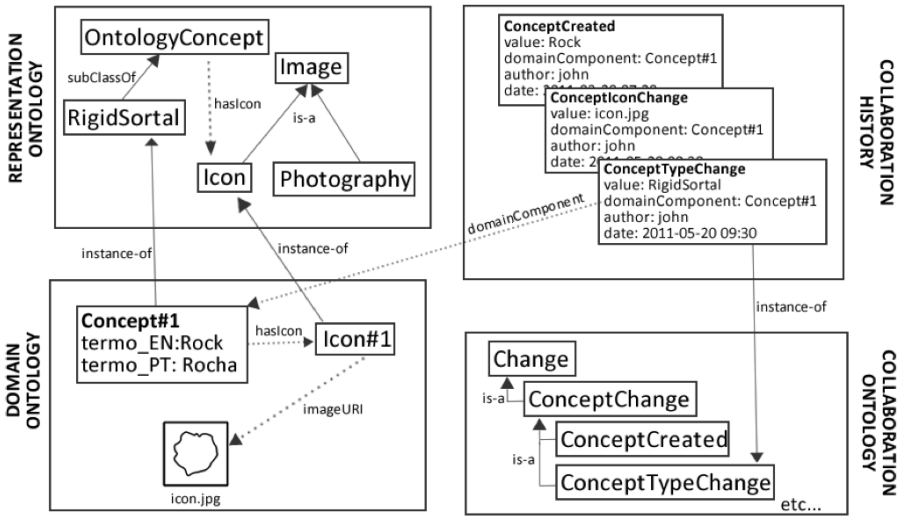


Figure 1. The collaboration structure

Currently, this approach covers *Endurants* objects in building models. But it is important, for Geology interpretation or natural phenomena, to represent the transformation of objects (*Endurants*) through the events.

4. Supporting Events from UFO-B

This work is an extension of the work presented in [Torres et al. 2011, Torres 2012], in which it was developed an environment to support collaborative construction and evolution of domain ontologies based on UFO-A. Now, we are concerned with the temporal aspects, i.e., with the representation of UFO-B events in that same environment. This section presents the modifications realized in the original meta-data ontologies ($R.O.$ and $C.O.$) used to specify the structure of the domain ontology components and collaboration

episodes. These modifications were realized in order to represent the temporal aspects in the domain ontologies created for this system. The first change made was to organize the meta-data ontologies to reflect the taxonomic structure of the foundational ontology UFO. Figure 2 depicts the main differences between the original and the new taxonomy in the $R.O.$.

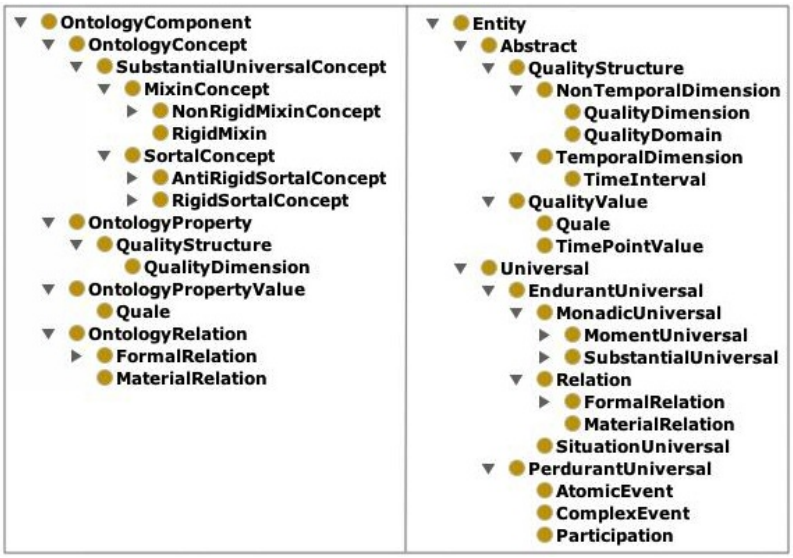


Figure 2. On the left, a fragment of the former $R.O.$. On the right, the updates performed

We renamed some terms of this meta-data ontology to conform to terminology used in UFO, and thus avoid possible confusion between the terms used in this work and in the UFO. For example, the term *OntologyComponent* was the concept from which all other concepts were specialized. In UFO, the concept that is the *top concept*, i.e., the concept from which all other concepts were specialized is the concept named *Entity*, so we renamed the term *OntologyComponent* to *Entity*. We have also made changes in the taxonomy of this meta-data ontology. For example, the *Relations* were structured as expected in the UFO, i.e., as an *EndurantUniversal*. There were many changes in nomenclature and taxonomy to match the terms used with those predicted in the UFO.

In order to organize the changes made, we will first present the theoretical concepts of UFO-B, and then show the changes made to represent the concepts in the approach proposed by Torres. Below, we explain some theoretical concepts from UFO-B along with the modifications made in the meta-data ontology to support the representation of them.

Mereological Structure: Events are examples of entities that obey the so-called Extensional Mereology, we have that: i) No event is part of itself; ii) If event X is part of event Y then event Y is not part of event X; iii) If event X is part of event Y and event Y is part of event Z then event X is part of event Z; iv) If event Y is part of event X then there is an event Z disjoint from Y which is also part of X; v) Two events are the same if and only if they are composed of exactly the same parts. Figure 3.A depicts the mereology structure in the $R.O.$.

Ontological Dependence: The Events from UFO-B are ontologically dependent on a relationship of participation with any object from UFO, i.e., an event exists only

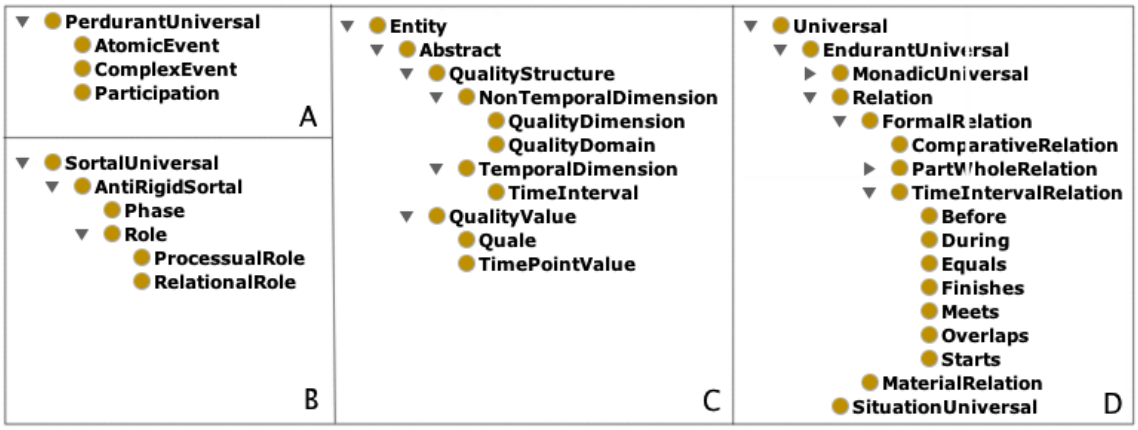


Figure 3. A) the mereologic structure of events and the Participation concept; B) the role differentiation in the UFO-B; C) the temporal qualities; D) the time interval relations

if at least one object is participating on it. The UFO-B provides the use of a special construct to establish a participation relation with one *object*, the construct is called *Participation* and it is depicted in Figure 3.A.

Role Diferentiation: When an object is participating in an event, he is playing a role in this event, which is not the same role that this object could play out of this event. Then we specialize the UFO-A *Role* concept in two types of *roles*: Those who are existentially dependent on some *Endurant* and those who are existentially dependent on some *Perdurant*. We call, respectively: *Relational Role* and *Processual Role*. Figure 3.B depicts this distinction in the $R.O.$.

Temporal Qualities: Events can be bearer of qualities, for example, a football game can be disputed, a conversation can be boring, and depositional process can be slow. Every event has a common property: its duration. Every event is framed by a time interval. And a time interval is associated with a temporal structure, which is analogous with a quality structure. An event is framed by a time interval T if we have TP_b and TP_e, that are quality values in a given temporal structure and, TP_b is the begin time point, and TP_e is the end time point. Figure 3.C depicts the changes realizes in the $R.O.$ in order to represent the time structures.

Time Interval Relations: Events may have relationship between then. We have a special kind of relationship between events which can be used to establish relationships of a partial or total ordering in their occurrences, for this we use de called Allen's Relations. These relations are added in the $R.O.$ and it is depicted in the Figure 3.D.

Change in the *State of Affairs*: Events can change the world by changing the *State of Affairs*. A *Situation* is defined in the UFO-A and it represents a particular *state of affairs*, i.e., a situation can be seen as a portion of reality that can be understood as a whole. An event takes one situation to another: from a pre-situation to a post-situation, i.e., an event has a pre-situation relationship with a particular situation, and a post-situation relationship with another particular situation different from first. We can define a Situation as a set consisting of several objects (including another situations). The inclusion of the Situation construct in the $R.O.$ are depicted in Figure 3.D.

The modifications in the $C.O.$ are still not completed in the date that this paper was written. Then, this discussion will be taken at a future opportunity.

5. Conclusions and Future Work

This work presented some modifications performed in the Torres' work to support the representation of UFO-B events in an ontology collaborative construction system. The objective of this project, in the long run, is to support the reasoning of geological interpretation of depositional process based on described characteristics of sedimentary rocks. We expect that a well founded domain ontology, built with the support of a collaborative tool, would express the geological concepts in a more precise way, therefore supporting useful interpretations.

The meta-ontologies introduced in the Torres' work were developed to provide a basis for the collaborative construction of language-independent domain ontologies, and were based on the constructs from UFO-A. Now, this work is expanding this basis, improving the meta-ontologies and supporting the representation of events and temporal aspects inherited from UFO-B. Currently this work is in constant development.

6. Acknowledgements

The scholarships in this project are funding by the program PETROBRAS PFRH 17 and CNPq. Material funding was supported by ENDEEPER Knowledge Systems.

References

Carbonera, J. L. (2012). Raciocínio sobre conhecimento visual: Um estudo em estratigrafia sedimentar. Master's thesis, Universidade Federal do Rio Grande do Sul.

Carbonera, J. L., Abel, M., Scherer, C. M. S., and Bernardes, A. K. (2011). Reasoning over visual knowledge. In *Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies*. Ontobras/Most.

Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. PhD thesis, CTIT, Centre for Telematics and Information Technology, Enschede.

Guizzardi, G., de Almeida Falbo, R., and Guizzardi, R. S. S. (2008a). Grounding software domain ontologies in the unified foundational ontology (ufo): The case of the ode software process ontology. In *CIbSE*, pages 127–140.

Guizzardi, G., Falbo, R., and Guizzardi, R. S. S. (2008b). A importância de ontologias de fundamentação para a engenharia de ontologias de domínio: o caso do domínio de processos de software. *Revista IEEE América Latina*, v. 6(n.3):p. 244–251.

Torres, G. M. (2012). Construção colaborativa de ontologias para domínios visuais utilizando fundamentação ontológica. Master's thesis, Universidade Federal do Rio Grande do Sul. No Prelo.

Torres, G. M., Lorenzatti, A., Rey, V., da Rocha, R. P., and Abel, M. (2011). Collaborative construction of visual domain ontologies using metadata based on foundational ontologies. In *Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies*. Ontobras/Most.

Aplicação de um Metamodelo de Contexto a uma Tarefa de Investigação Policial

Lucas A. de Oliveira, Rui A. R. B. Figueira, Expedito C. Lopes

Mestrado em Sistemas e Computação – Universidade de Salvador (UNIFACS) – Salvador – BA - Brasil

{ruialexandrefigueira,lucas.amorim969}@gmail.com,
expedito.lopes@pro.unifacs.br

Abstract. *Ontologies define a specific vocabulary to describe a certain reality, while Context is knowledge that helps to identify what is relevant in a given situation. Among the classifications, there are task and domain ontologies, which combined, produce a vocabulary that describes concepts related to a set of applications that work by performing a particular task in a related field. Currently, few studies have addressed issues involving ontologies and is not a trivial task its integration with context. This article presents partial results of a study whose objective is to apply a context´s metamodel to a Police Investigation Task.*

Resumo. *Ontologias definem um vocabulário específico para descrever certa realidade, enquanto Contexto é um conhecimento que ajuda a identificar o que é relevante em determinada situação. Dentre as classificações, existem ontologias de tarefa e de domínio, que combinadas, produzem um vocabulário que descreve conceitos relativos a um conjunto de aplicações que atuam realizando uma determinada tarefa em um respectivo domínio. Até o presente, poucos trabalhos têm abordado temáticas envolvendo Ontologias de Tarefa e, não é trivial sua integração com contexto. Este artigo apresenta os resultados parciais de uma pesquisa cujo objetivo é aplicar um metamodelo de contexto a uma Tarefa de Investigação Policial.*

1. Introdução

Uma Ontologia envolve a descrição de conceitos, suas propriedades, relações e suas restrições referentes a um determinado conhecimento, que são senso comum para um grupo de especialistas nesse conhecimento. Assim, a comunicação, integração, busca, armazenamento e representação do conhecimento são facilitados (O'LEARY, 1998). Quanto às classificações, há diversas propostas para as Ontologias, como a de Guarino (1998), que as define como: a) *Ontologias de Fundamentação (ou Topo)*, que abrangem conceitos muito genéricos, como evento, tempo, problema, etc., b) *Ontologias de Domínio*, as quais descrevem os conceitos de um domínio específico, como Medicina, Polícia, Computação, etc., c) *Ontologias de Tarefa*, que possuem o vocabulário de uma tarefa genérica, como comprar, vender, investigar, etc., e d) *Ontologias de Aplicação*, que são resultado de uma Ontologia de Domínio com uma Ontologia de Tarefa, empregadas numa aplicação em particular.

Um modelo para integração entre Ontologias de Domínio e Tarefa foi proposto por (MARTINS, FALBO, 2008), considerando que esta integração não apenas serve para descrever o conhecimento de uma aplicação em particular, mas também de uma classe de aplicações, designando assim uma *Ontologia de Classes de Aplicação*.

Por sua vez, contexto é o conhecimento que ajuda a identificar o que é ou não relevante em um dado momento e lugar. O contexto não é uma entidade autônoma, mas existe, apenas, quando relacionado a alguma entidade. Caracteriza-se por ser dinâmico, e depende da tarefa atual e do agente que a executa (VIEIRA, 2008).

Na literatura existente, percebe-se que a maioria dos trabalhos sobre ontologias tratam de Ontologias de Domínio, sendo bastante reduzida a quantidade de trabalhos que tratam Ontologias de Tarefas ou acerca da integração de ambos os temas.

Por outro lado, vários trabalhos têm mostrado que a inclusão de Contexto na modelagem traz inúmeras vantagens, tais como representar dinâmica de contextos e criar aplicações mais adaptativas e adequadas às necessidades dos usuários (VIEIRA, 2008).

Este trabalho tem por objetivo aplicar um metamodelo de contexto em tarefas de investigação policial, que é resultado da integração de conceitos de tarefa genérica (investigação) com conceitos de um domínio particular (polícia).

O restante deste trabalho está estruturado assim: a seção 2 contém os principais conceitos relativos a contexto. A terceira seção, está dividida em 3 partes: apresentação do metamodelo de contexto proposto por Vieira (2008); apresentação da tarefa *Investigação Policial*, que foi construída com base no perfil UML proposto por Martins (2009); e a integração entre o metamodelo de contexto e a tarefa em questão, gerando um diagrama contendo uma tarefa que considera conhecimento contextual, o que não é trivial e ainda é pouco utilizado. Por fim, a seção 4 contém conclusões, além de sugerir possíveis trabalhos futuros.

2. Conceitos Fundamentais

Nesta seção são apresentados os principais conceitos necessários ao entendimento deste trabalho.

2.1. Contexto

Contexto pode ser definido como as circunstâncias em que ocorre um evento. Com relação à comunicação entre pessoas, o contexto revela a história de tudo que ocorreu num determinado tempo, o estágio de conhecimento dos agentes participantes bem como um conjunto de expectativas existentes naquele momento (BRÉZILLON, 1999).

Dey e Abowd (2001) afirmam que contexto é qualquer informação que caracteriza a situação de uma entidade, em que uma entidade é um lugar, pessoa ou objeto considerado relevante para a interação entre o usuário e a aplicação. Necessariamente, o contexto precisa estar associado a alguma outra entidade tal como um agente, interação ou tarefa para existir. Uma tarefa descreve uma atividade por meio da especialização de conceitos introduzidos previamente. Agentes, de modo geral constituem elementos autônomos que representam, manipulam e trocam conhecimentos e informações.

Um elemento contextual representa um tipo de informação que pode ser conhecida, codificada e também representada antecipadamente; além disso, o elemento contextual é qualquer dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio (VIEIRA, 2008).

2.2. Foco

Brézillon (2007) define *foco* como sendo um passo importante na execução de uma tarefa ou em um processo de tomada de decisão, possibilitando estabelecer quais elementos contextuais devem ser instanciados e usados para constituir um contexto.

Segundo Brèzillon (2007), *foco* representa a associação de uma *tarefa* a um *agente*, o qual recebe um *papel* para executar esta tarefa. Ao realizar alguma ação, o foco atual da pessoa consiste na execução do passo específico que se relaciona a finalização de alguma tarefa.

Como exemplo, o foco "MédicoRealizaDiagnóstico", representa uma tarefa "realizar diagnóstico" para um agente "médico" no papel "analista".

2.3. Entidades Contextuais

Entidades Contextuais representam as entidades do modelo da aplicação que devem ser consideradas para fins de manipulação das informações contextuais (VIEIRA, 2008).

Por sua vez, uma entidade contextual pode ser caracterizada por meio de elemento contextual identificado a partir de um conjunto de relacionamentos e atributos associados a uma entidade contextual. Os elementos contextuais podem ser detectados por meio de um conjunto de relacionamentos e atributos associados à entidade que o contém (VIEIRA, 2008).

Considerando o domínio de missões acadêmicas, tem-se *Aluno* e *Missão* como exemplos de entidades contextuais; e *nívelEscolaridade* e *orientador* como exemplos de elementos contextuais presentes em *Aluno,* ou *localRealização* e *duração* presentes na entidade contextual *Missão*.

3. Aplicação de um Metamodelo de Contexto a uma Tarefa de Investigação Policial

Nesta seção, a tarefa *Investigação Policial,* que foi elaborada com base no perfil UML proposto por Martins (2009), será combinada ao metamodelo de contexto proposto por Vieira (2008), resultando num diagrama que representa conhecimento de tarefa com contexto.

3.1. Metamodelo de Contexto

Em Vieira (2008), um metamodelo de contexto é apresentado, o qual é independente do domínio e permite modelagem de contexto em diferentes aplicações, com aspectos estruturais e comportamentais envolvidos no uso e gerenciamento de contexto de forma integrada. O metamodelo é apresentado na Figura 1, descrito em UML, onde se pode observar a existência de diversos conceitos, suas propriedades e relações.

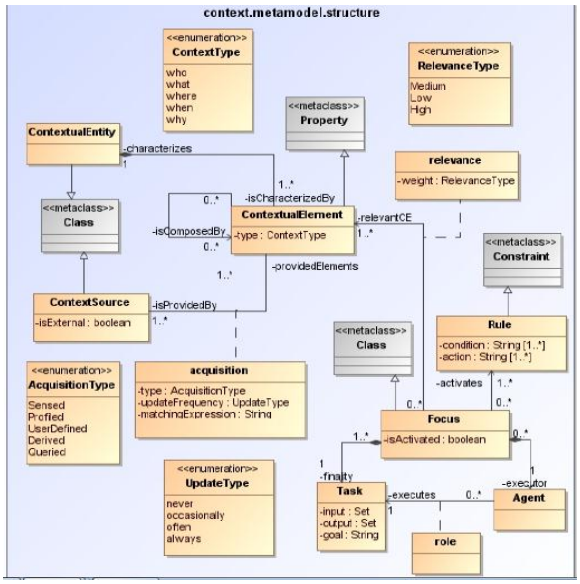


Figura 1. Estrutura do metamodelo de contexto. Fonte: Vieira (2008).

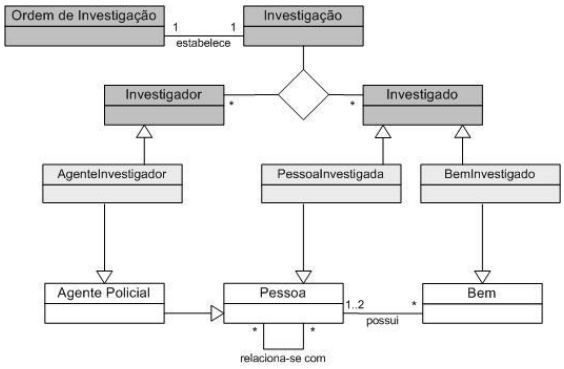


Figura 2. Diagrama da tarefa Investigação Policial.

As classes Tarefa (*Task*), Foco (*Focus*), Agente (*Agent*), Papel (*role*), Entidade Contextual (*ContextualEntity*) e Elemento Contextual (*ContextualElement*) são usadas neste trabalho, uma vez que são importantes no uso do metamodelo na tarefa de investigação. As demais classes não são utilizadas, pois não são necessárias para o objetivo supracitado.

3.2. Tarefa Investigação Policial

Com base no perfil UML proposto por Martins (2009), elaborou-se um diagrama que contém conceitos da tarefa *Investigação Policial*, que pode ser visualizado na Figura 2.

Ela representa o modelo estrutural resultante da integração dos conceitos de tarefa genérica (*Investigação*) com os conceitos do domínio (Polícia). Ela contém os termos que são essenciais a qualquer aplicação de investigação policial, mas deixa de lado conceitos e restrições mais específicos de uma aplicação em particular, que é competência de uma ontologia de aplicação. Para facilitar a compreensão, os elementos que representam conceitos da tarefa estão com o fundo cinza escuro, enquanto conceitos do domínio estão com o fundo branco. Vale ressaltar que os conceitos da tarefa *Investigação* foram desenvolvidos de forma independente do domínio, o que torna mais fácil a sua integração com o conhecimento de domínio. Assim, esse conhecimento da tarefa de investigação pode ser portável, por exemplo, para uma investigação médica, onde o médico investiga uma doença.

Os termos usados na Figura 2 são genéricos e representam os papéis que as entidades do domínio exercerão ao executar a tarefa. A integração se dá justamente ao associar quais os conceitos do domínio desempenharão os respectivos papeis dentro da tarefa. Os elementos dessa integração estão apresentados com o fundo cinza claro. Observando-se a Figura 2, é possível perceber que A*genteInvestigador* é uma especialização de *Agente Policial*. Isso significa que específicos elementos do conceito Agente Policial (motorista, escriturário, investigador, etc), estão representados no

conceito Agente Investigador (investigadores). Consequentemente *Agente Investigador* é uma especialização do conceito *Investigador* (representando tarefas que fazem investigação). Considerando a tarefa de investigação o conceito investigado pode ser representado por mais de um conceito presente no domínio Policial usado neste trabalho: *Pessoa* e *Bem*.

3.3. Aplicação do Metamodelo à Tarefa de Investigação Policial

Tomando por base o metamodelo e o perfil UML expostos nas subseções acima, construiu-se um diagrama contendo a tarefa *Investigação Policial* que leva em consideração o conhecimento de contexto. Ele é apresentado na Figura 3.

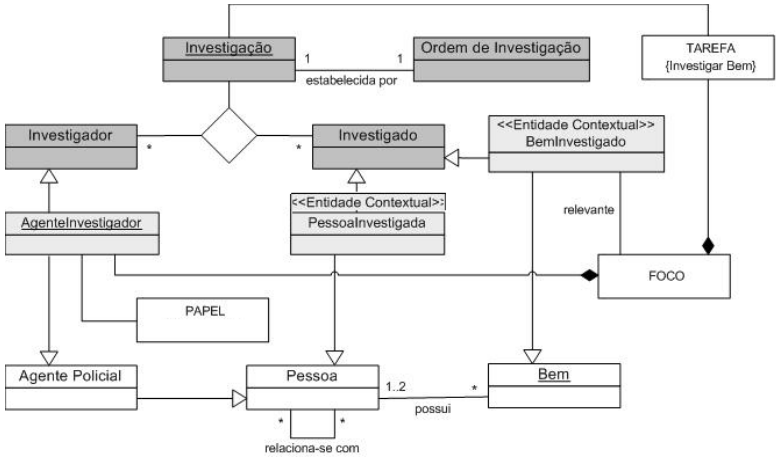


Figura 3. Diagrama de tarefa com contexto a partir de um bem investigado.

Neste primeiro momento, a entidade contextual *BemInvestigado* foi representada no diagrama sem atributos e elementos contextuais (atributos tal como *descriçãoBem* e elementos contextuais, como por exemplo *proprietárioAtual*, serão considerados posteriormente). *BemInvestigado* é relevante para o foco onde um agente investigador faz uma investigação a partir de um bem de uma pessoa investigada. Como foco é uma tarefa executada por um agente (pessoa ou software) considerando elementos contextuais relevantes, o conceito *FOCO* é uma composição de *AgenteInvestigador* com *Tarefa,* associado a *BemInvestigado*. Com base no diagrama acima importa considerar que o conceito *AgenteInvestigador* está associado ao conceito *PAPEL* significando que um agente que faz investigação representa uma função (Investigador) associada ao foco.

A integração entre o metamodelo e o perfil UML apresentado se dá através da associação entre *TAREFA* e *Investigação*; Além de *PAPEL* e *AgenteInvestigador*; e por fim, *FOCO* e *BemInvestigado*, conforme Figura 3.

Se considerarmos outro foco, por exemplo, um agente investigador vai investigar uma pessoa, a tarefa ainda é a mesma e o agente exerce o mesmo papel, mas, neste caso, o conceito *FOCO* deve ser associado à entidade contextual *PessoaInvestigada* e um possível elemento contextual seria *Alcunha*. Esta situação é representada Figura 4.

4. Conclusões e Trabalhos Futuros

Percebendo que há poucos trabalhos abordando ontologias de tarefa e suas representações, procurou-se estudar propostas relacionadas a este tema. Além disso, o

uso de contexto tem proporcionado boas vantagens aos sistemas, pois torna suas aplicações mais adaptativas e adequadas às necessidades dos usuários.

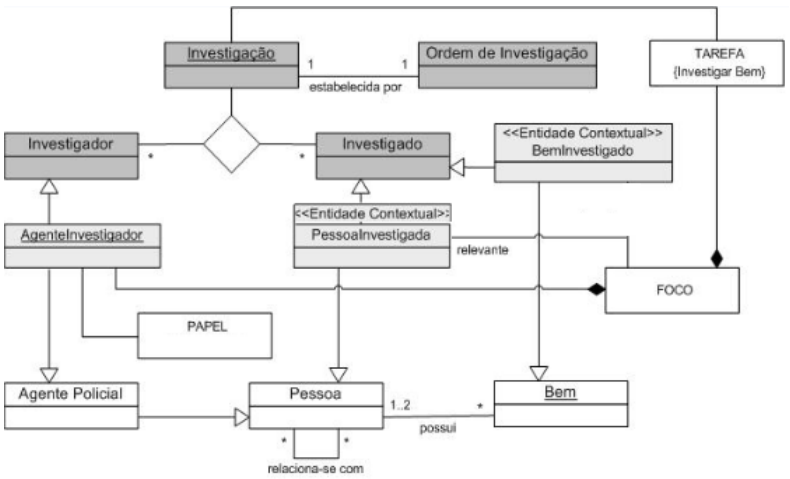


Figura 4. Diagrama de tarefa com contexto a partir de uma pessoa investigada.

Neste trabalho foi aplicado um metamodelo de contexto já consolidado a uma importante tarefa do domínio Policial, a Investigação, tendo como base a representação de ontologia de tarefa usando perfil UML proposta por Martins (2009).

Como possíveis trabalhos futuros, sugere-se: (i) realizar a representação da ontologia da tarefa Investigação Policial no Protégé (http://protege.stanford.edu), a fim de obter seu correspondente XML ou OWL; (ii) criar regras de inferência para em seguida gerar os axiomas da ontologia; e (iii) expandir o modelo, adicionando novas tarefas e classes, tais como *Cargo*, *Atribuição*, *Endereço*, etc.

Referências

Brézillon, P. (1999) "Context in Artificial Intelligence: IA Survey of the Literature", Computer&Artificial Intelligence, v. 18, pp. 321-340.

Dey, A. K.; Abowd, G. D. (2001) "A Conceptual Framework and a Tookit for Supporting the Rapid Prototyping of Context-Aware Applications", Human-Computer Interaction (HCI) Journal, v. 16, n. 2-4, pp. 97-166.

Guarino, N. (1998) "Formal Ontology and Information Systems", In: Formal Ontologies in Information Systems, N. Guarino (Ed.), IOS Press, pp. 3-15.

Martins, A. F. (2009) "Construção de Ontologias de Tarefas e sua Reutilização na Engenharia de Requisitos", Tese de Mestrado, Espírito Santo: UFES.

Martins, A. F., Falbo, R. A. (2008) "Models for Representing Task Ontologies". Proceedings of the 3rd Workshop on Ontologies and their Aplications (Wonto'2008), Salvador, Brasil.

O'Leary, D. E. (1998) "Using AI in Knowledge Management: Knowledge Bases and Ontologies", IEEE Intelligent Systems, v. 13, n. 3, p. 34-39.

Vieira, V. (2008) "CEManTIKA: Um Framework Independente de Domínio para Projetar Sistemas Sensíveis ao Contexto", Tese de Doutorado, Recife: UFPE.