Ontology Issues and Applications
Guest Editors' Introduction

Fred Freitas
Centro de Informática
Universidade Federal de Pernambuco
Caixa Postal 7851
Cidade Universitária - 50732-970
Recife, PE, Brazil
fred@cin.ufpe.br

Heiner Stuckenschmidt
Department of Mathematics
and Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1081a - 1081 HV
Amsterdam, The Netherlands
heiner@cs.vu.nl

Natalya F. Noy
Stanford Medical Informatics
251 Campus Drive
Stanford,
CA 94305-5479,
USA
noy@stanford.edu

1. INTRODUCTION

After being studied by philosophers for twenty three centuries, "ontologies" have recently become a new buzzword in computer science. The topic is currently receiving special attention not only from an active community of researchers from many areas of informatics, but also from the industry, which is providing increasing budgets and investments to develop this technology and to enhance its applicability in business settings.

In computer science terms, an ontology comprises a set of definitions of concepts, properties, relations, constraints, axioms, processes and events that describe a certain domain or universe of discourse. By providing this body of definitions about a domain, an ontology enables applications and software agents to use the precise, clear, formal semantics to process the information described by the ontology and to use this information in intelligent applications.

Many areas of computer-based applications are now taking advantage of ontologies: knowledge management, electronic commerce, tutoring and geographic systems, e-government, among many other application areas.

Ontology research and practice has recently received a very strong boost from the idea of the Semantic Web, popularized by Tim Bernes-Lee—the key inventor of the today's World Wide Web. Berners-Lee defines the Semantic Web as a Web in which the software will "understand" and process data from the pages, according to the context of this data [46]. Ontologies constitute the backbone of the Semantic Web, as they are responsible for providing this context. A Web page then instantiates an ontology about the domain that this page refers to, and a software agent that handles the page can use the precise semantics of these definitions to process the information.

While ontology researchers have made many advances in recent years, many key challenges still must be addressed, including topics such as ontology interoperability and diversity, engineering methodologies, Semantic Web standard and practices, and other challenging issues.

Given the huge variety of readers of this Special Issue, which span from undergraduate students to senior researchers and IT practitioners, this extended editorial aims at providing the readers a brief description of the field, giving a flavor of ontologies' rich history and application possibilities, as well as the interesting research that is taking place and its challenges to overcome. The end of the editorial brings an outline of the articles of this Special Issue, which by no means reflect all the research topics and applications of the field in growth, but rather sketch a portrait of part of the ontology research that is

taking place in Brazil now. Since new applications and opportunities are flourishing, we hope that this Special Issue and this text motivate the readers, and especially those from the Brazilian research and industrial community, to join the large group of researchers who devote their efforts to the development of ontology technology.

2. ONTOLOGY DEFINITIONS

It was from Tom Gruber, a lead researcher of the ontology project Knowledge Sharing Effort (KSE) [19], that came the most often cited definition of an ontology: "an ontology is an explicit specification of a conceptualization" [18]. His definition focuses on the point that ontologies are declarative structures ("explicit") and must depict concepts from a certain domain or universe of discourse ("conceptualization"). An interesting point in that definition is its abstraction: it does not commit unnecessarily to any way in which the specification is actually carried out.

This definition, however, does not make explicit, some of the key features and benefits of ontologies, which were already present in KSE's work and in the field as a whole: applicability of ontologies in deductive systems, in agent communication and their role in standardizing knowledge. In order to comprise these features, Gruber's definition was later refined by several other researchers to "an ontology is a *formal* explicit specification of a *shared* conceptualization" [21,49]. By being defined in a formal language, an ontology could be handled by a reasoning engine. By being shared, an ontology can represent a consensus about the area of knowledge that it refers to, or, at least, gather definitions that will be shared with the same semantics by intelligent agents engaged in communication.

A number of other definitions appeared in the meantime, many of them being more concrete, in the sense that they are more precise about what constitutes an ontology, rather than what an ontology itself is. For example, Robert Neches had stated that "an ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary" [38].

3. A BIT OF HISTORY

The term "ontology" is rooted on Philosophy. It could be considered a twin of Logic, as both were created by Aristotle. While trying to mimic the mechanics of human reasoning, this extraordinary Greek realised that, in order to obtain sound mechanical inferences, it was necessary to supply the deduction process with a minimum volume of knowledge into which the diverse objects of the world, either abstract (like ideas) or concrete (like a cat), could be classified. The classified objects would inherit all the predications associated with its related class. Aristotle also created a structure to build class hierarchies, which is based on ten top classes defined by him, known as *Categories*. Though incipient, this pioneering work was anticipating and paving the way for the birth of many branches of modern computer science and artificial intelligence, such as common-sense reasoning, ontologies and object-orientation, to name but a few. From the XVII century until its success in informatics, this term would be studied in Philosophy as a branch of metaphysics, focused on the distinction amongst the objects of the world, their relations and dependencies.

The use of ontologies in computer science and AI began after the rise of expert systems in the mid-'80s. As declarative structures that could be used for mechanical reasoning, ontologies were natural candidates as components in knowledge-based systems. Projects such as Cyc [29] and Sowa's top ontology [49] were trying to endow knowledge based-systems with the ability of common-sense reasoning. Common-sense reasoning is the type of inference present in our daily life, in which we take into account loose facts like 'all animals are born and die' and 'things left in the air usually fall' to make sound inferences. The ontologies of this AI research phase were typically what we call now top-level ontologies: large, comprehensive, aiming at including definitions about everything. Although common-sense reasoning proved a tough task to accomplish and is still a major long-term goal of AI research, the construction and use of ontologies became a new discipline in AI.

Ontologies grew in popularity when their focus became more restricted. Ontologies describing single domains, such as bibliography and microbiology, available in repositories such as the Ontolingua [18], started to support knowledge reuse among knowledge-based systems (KBSs). For these systems, the knowledge base construction represents the dearest investment, and, until the 90's, knowledge reuse was hampered for two main reasons.

Firstly, in early expert systems, knowledge was designed focusing on tasks rather than on domains, like in ontologies. For instance, the very first expert system Mycin [48], which diagnosed bacterial infections, had no explicit descriptions about the concepts and relations of the microbiological domain, like infections, organisms, processes, and other entities. Nowadays, if new users go about solving a new task in microbiology, they could benefit from off-the-shelf medical and biological ontologies that can play the role of a rich vocabulary for their systems, instead of starting from scratch.

Finally, there was the diversity of knowledge representation formalisms, in which ontology editors came up with good solutions, as can be seen in subsection 4.2. Nowadays, dealing with diversity of expressiveness

(formalisms) and contents (perspectives, terminology, points of view, meaning) turn out to be recurrent and challenging research issues for ontology employment, and also a good source of ontology usage, e.g. for problems like information integration of heterogeneous databases and systems, as described in section 4.3.

4. ONTOLOGY RESEARCH

In this section, we introduce some of the more active, but by no means all, areas of ontology research. We start by discussing advances in ontology engineering and evaluation and the tools that enable ontology development. Using ontologies for information integration from multiple sources and reconciling of multiple ontologies is probably one of the more difficult and more active research areas. As we mentioned earlier, the idea of the Semantic Web provided a huge boost to ontology research and also introduced many new challenges.

4.1 ONTOLOGY ENGINEERING AND EVALUATION

At first, ontology development resembled more artwork than engineering, with each team adopting its own set of principles, criteria and design phases [17]. Today, in a clear sign of progress, sound engineering methodologies to support ontology development are emerging and are being adopted. Indeed, the process of ontology construction is shifting from these ad-hoc efforts to a rigorous engineering discipline. This shift is still under way, and ontology engineering methodologies constitute an active area of research nowadays.

To a certain extent, ontology building methodologies reminds of system analysis. They provide guidance to developers, have similar iterative phases and are also defensive by nature [9]: there is no single engineering methodology that leads to correct ontologies, but they help users to avoid common mistakes that would certainly be harmful.

In many of the engineering methodologies, the usual phases of ontology construction are *specification, conceptualization, implementation* and *evaluation.* The specification phase aims at defining the purpose and scope of the new ontology. During conceptualization, the ontology is populated with the definitions. In some methodologies, the phases of evaluation and implementation are merged into a single one, once here the ontology is coded in a knowledge representation formalism (implementation) and tested against the requirements defined in the first phase. To have an in-depth, extensive analysis of ontology engineering methodologies, the interested reader should rely on the book written by Gómez-Pérez, Fernandez and Corcho [17].

Once an ontology is developed, we must evaluate it from many perspectives: how well does it reflect our original goals, how well it is suited to our potential application, how well does it correspond to formal principles of ontology design. Thus, the phase of *ontology evaluation* has become an independent sub area of ontology research. One way to evaluate an ontology is to consider whether its structure conforms to principles grounded on centuries of Philosophy work. In this type of evaluation, we check the validity of some constraints against concepts' formal metaproperties, such as rigidity, identity, unity, parthood and dependencies. A practical example shall explain better [22]: the linguistic ontology WordNet [36] says that "physical-object is-a amount-of-matter", while top ontology Pangloss [52] states the opposite. Which one is correct, if any? Checking the metaproperty of unity of both concepts, one could draw to the conclusion that both perspectives are wrong: an amount of matter can not be viewed as a "whole" while objects must be. Thus, inheritance cannot hold between these concepts in either direction, since one of the constraints for unity says that unity criteria must match for inheritance. OntoClean [22] is currently the leading methodology for ontology evaluation. It is under use in corporations and research labs for checking ontology consistency and assisting reengineering of ontologies.

4.2 DIVERSITY OF FORMALISMS: ONTOLOGY EDITORS

One of the factors that hindered ontology reuse was the diversity of formalisms used to represent ontologies (e.g., semantic networks, frames and description logics definitions). Although these formalisms were often semantically close to each other, there were no easy, clean, formal transformations available among them or among its various representation languages (such as Prolog, F-logic, RDF and XML) to support knowledge reuse.

An important step for ontologies to be accepted as a large-scale technology for conceptual modelling and knowledge reuse was the deployment of easy-to-use graphical ontology editors. Besides being fairly user-friendly, these tools are largely adopted because they hide the complexities of formalisms from the user, also allowing ontologies created graphically to be automatically translated into a number of formalisms and representation languages, including the ones from the Semantic Web. Among the several ontology editors that appeared, Protégé [43], WebODE [3] and OntoStudio (past OntoEdit) [45] can be mentioned as some of the most popular editors. One main aspect shared by them seems to justify their popularity: a flexible knowledge model that simplifies the operations needed to import/export ontologies from/to various representation languages and their formalisms.

Other requirements that make ontology editors successful stem from software engineering, such as extensibility and plugability. Protégé, for instance, provides

a software structure to ease the addition of components (plug-ins) built by the increasing number of users. Plugins for Protégé, WebODE and OntoStudio encompass, for instance, reasoning services that enables the use of reasoning services as an integral part of an ontology-development process, providing facilities for ontology "debugging." In recent years, various inference engines, verifiers, graphical interfaces and database import/export components have become available and are employed worldwide.

4.3. DIVERSITY OF ONTOLOGIES: ONTOLOGY MERGING, ALIGNMENT AND ONTOLOGY-BASED INFORMATION INTEGRATION

An important application area of ontologies is the integration of existing systems/databases. The ability to exchange information at run time, also known as interoperability, is an important topic. The attempt to provide interoperability suffers from problems similar to those associated with the communication amongst different information communities. The important difference is that the actors are not persons able to perform abstraction and common sense reasoning about the meaning of terms, but machines. In order to enable machines to understand each other, we also have to explicate the context of each system, but on a much higher level of formality in order to make it machine understandable. Ontologies are often used as interlinguas for providing interoperability [54]: they serve as a common format for data interchange. Each system that wants to interoperate with other systems has to transfer its information into this common framework.

In existing ontology-based integration approaches like SIMS [2], TSIMMIS [15], PICSEL [30], Ontobroker [8], SHOE [25] and OBSERVER [35], ontologies are used for the explicit description of the information-source semantics. But there are different ways of how to employ the ontologies. In general, three different directions can be identified: single-ontology approaches, multiple-ontology approaches and hybrid approaches.
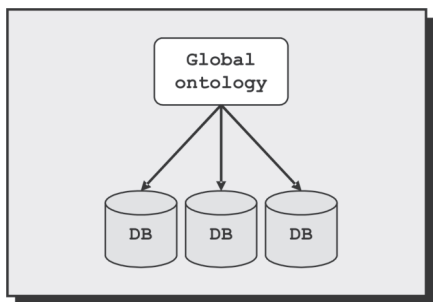
Single-ontology approaches use one global ontology providing a shared vocabulary for the specification of the semantics (see Figure 1). All information sources are related to the one global ontology. A prominent approach of this kind of ontology integration is SIMS. SIMS model of the application domain includes a hierarchical terminological knowledge base with nodes representing objects, actions and states. An independent model of each information source must be described for this system by relating the objects of each source to the global domain model. The relationships clarify the semantics of the source objects and help to find semantically corresponding objects. Single-ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view of a domain. But if one information source has a different view of a domain, e.g. by providing another level of granularity, finding the minimal ontology commitment [17] becomes a difficult task. For example, if two information sources provide product specifications but refer to absolute heterogeneous product catalogues which categorize the products, the development of a global ontology which combines the different product catalogues becomes very difficult. Information sources with reference to similar product catalogues are much easier to integrate. Also, single-ontology approaches are susceptible to changes in the information sources, which can affect the conceptualization of the domain represented in the ontology. Depending on the nature of the changes in one information source it can imply changes in the global ontology and in the mappings to the other information sources. These disadvantages led to the development of multiple-ontology approaches.
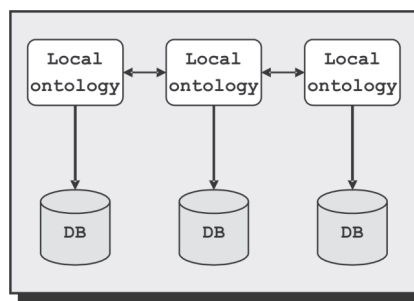

Figure 2: Multiple-Ontology Approach

Figure 1: Single Ontology Approach

In multiple-ontology approaches, each information source is described by its own ontology (Figure 2). For example, in OBSERVER, the semantics of an information source is described by a separate ontology. In principle, the "source ontology" can be a combination of several other ontologies but it cannot be assumed that the

different "source ontologies" share the same vocabulary. At a first glance, the advantage of multiple-ontology approaches seems to be that no common and minimal ontology commitment about one global ontology is needed. Each source ontology could be developed without reference to the other sources or their ontologies; thus no common ontology with the agreement of all sources is needed. This ontology architecture can simplify the change, i.e. modifications in one information source or the adding and removing of sources. In reality, however, this model brings out one of the more difficult problems in ontology research: mapping between different ontologies by finding similarities and differences between them. We discuss this problem later in the section.

To overcome the drawbacks of the single- or multiple-ontology approaches, hybrid approaches were developed (Figure 3). Similar to multiple-ontology approaches the semantics of each source is described by its own ontology. But in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary. The shared vocabulary contains basic terms (the primitives) of a domain. In order to build complex terms of a source ontology the primitives are combined by some operators. Because each term of a source ontology is based on the primitives, the terms become easier to compare than in multiple-ontology approaches. Sometimes the shared vocabulary is also an ontology.
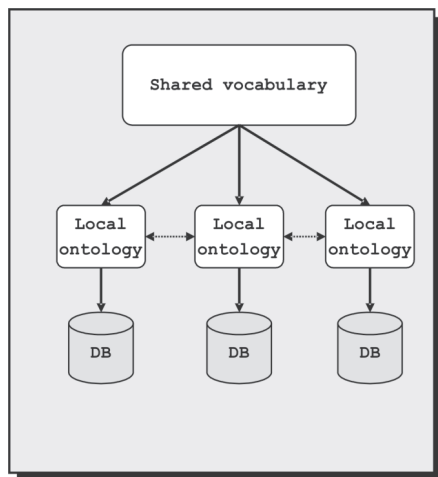

Figure 3: Hybrid Approach

The advantage of a hybrid approach is that new sources can easily be added without the need of modification in the mappings or in the shared vocabulary. It also supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of multiple-ontology approaches. The drawback of hybrid approaches, however, is that existing ontologies cannot be re-used easily, but have to be re-developed from scratch, because all source ontologies have to refer to the shared vocabulary.

As we mentioned earlier, finding correspondences between ontologies (*ontology mapping and alignment*) is one of the more difficult problems in ontology research, The issue arises not only in the context of information integration, but also more generally, whenever two ontologies with overlapping context need to be used in a single application or by a single software agent. In the ideal world, there will be standard ontologies describing models of different domains: one ontology for each area of medicine, one for business processes, one for travel applications, and so on. However, not only this is not the case today and multiple ontologies covering the same domains exist, but also the situation is likely to get worse in the future: as more ontologies are developed, there will be more ontologies with similar or overlapping content. It is unreasonable to expect that people will agree on a small set of ontologies with little or no overlap. Reasons range from practical (different applications require different views of a domain) to institutional and social (an ontology developed elsewhere could not be as good as the one we will develop ourselves). However, applications that use different ontologies for describing their domains, still need to interoperate. Therefore, we need to find correspondences between different ontologies. Given two ontologies, we need to be able to see what the similarities and differences are, and to express these correspondences (a mapping between the ontologies) in a machine-processable way. In this section, we discuss tools that help users identify the correspondences between ontologies. Our review here is necessarily brief. For a comprehensive survey of ontology-mapping tools, we refer the reader to an excellent survey by Kalfoglou and Schorlemmer [27]. In this subsection, we focus on the types of specific differences that can exist, and on different ways of specifying them.

The first class of tools for ontology mapping deals with the case where the two ontologies to be mapped share a common reference ontology, as shown in Figure 4(a). Several upper ontologies, such as SUMO [40] and DOLCE [14] are developed specifically for the goal of facilitating knowledge sharing. Grüninger and Kopena propose an approach to ontology integration that is based specifically on the idea of a shared interlingua, in which there are no direct mappings between the ontologies, but only to the interlingua, as deployed in Figure 4(b) [20].

When a shared ontology is not available, one can rely on discovery tools that help find direct mappings

between the ontologies, like in Figure 4(c). These tools use other types of information): lexical and structural information, user input, external resources, or prior matches. The tools developed by Hovy and colleagues [26] are probably the most representative of the tools using lexical information, such as concept names and definitions, their lexical structure, distance between strings, and so on.

The majority of tools for ontology mapping use some sort of structural or definitional information to discover new mappings. This information includes such elements as subclass–superclass relationships, domains and ranges of properties, analysis of the graph structure of the ontology, and so on. Some of the tools in this

category include QOM [11], Similarity Flooding [34], and the Prompt tools [42].

User input is another important source of information. Most researchers believe that completely automatic ontology mapping is beyond our reach and therefore some user interaction is required. This interaction may include seeding the mapping algorithm with initial set of matching pairs, verifying the matches that an algorithm produces, or configuring the specific matchers used [42,33,37].

Many external sources available in electronic form provide useful information for mapping discovery. The S-match algorithm [16], for instance, uses annotations from WordNet to help in finding mappings.
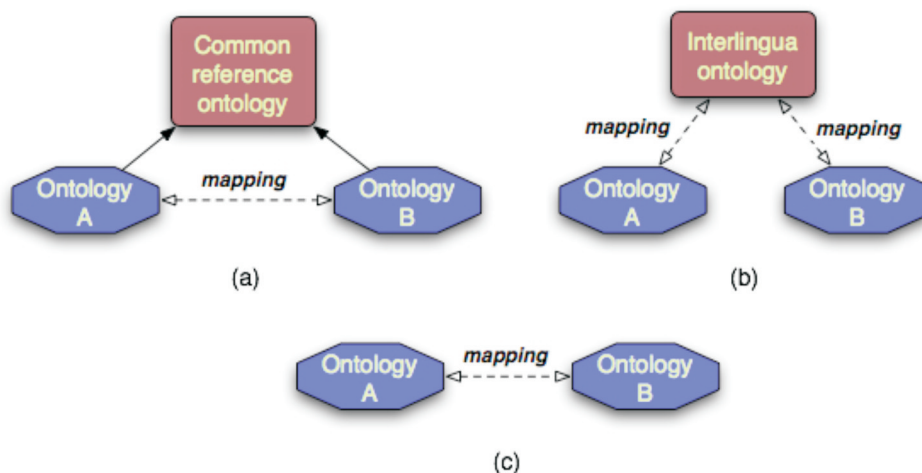

Figure 4: Different types of ontology mappings.

4.5 Semantic Web

We have already introduced the idea of the Semantic Web at the beginning of this article. To wit, today's Web is built primarily for consumption by humans. The idea of the Semantic Web is the Web of information represented in a formal way that can be processed, collected, aggregated, and analyzed automatically by machines. In order for software agents to "understand" this information, it will be described and annotated formally with ontologies. Ontologies themselves will be published on the (Semantic) Web, and will be easily accessible for others to reuse. The SW ontology languages, such as OWL (Ontology Web Language), provide specific mechanisms to facilitate such reuse. For instance, there are explicit ways to declare that one ontology imports another, that a class in one ontology is equivalent to a class in another ontology, and so on. With the World Wide Web Consortium (W3C) behind the Semantic Web, this field has already produced some

tangible advances crucial to ontology research. First and foremost, for the first time ever, there is a set of endorsed standard ontology languages: RDF(S) (Resource Description Framework) and OWL. These languages are W3C standards and provide the common language to represent ontologies and instance data, thus reducing the need for cross-formalism translation.

The Semantic Web also brings with it a long list of new and interesting challenges. First, one can think of the Semantic Web as one large web-scale knowledge-representation system. Therefore, many of the AI tools, including reasoners and inference engines, that are currently not very scalable will have to be rethought and redesigned to address the scalability issues of the Web.

Second, the Web brings with it the issues of security, reliability, and trust. How do you know which ontologies to reuse? How do you know that ontologies

you are reusing are "good" and reliable? How do you define policies for accessing your own ontologies and instance data that you make accessible on the Web? What do you do if an ontology you are relying on goes off line or is temporarily inaccessible?

Third, historically, reasoning in artificial intelligence and query-answering in databases has been about providing precise and complete answers. On the Web, however, precise or complete answers may never exist. Sites may or may not be accessible at any particular time, information may be only partially reliable, or contain only part of the data needed for an answer, and so on. We will need to learn how to deal with this fairly new realm of uncertainty and imprecision in inherently certain and precise processes (reasoning and querying).

All the research that we have described in this section, however, is going to be even more relevant in the context of the Semantic Web. With the wider use of ontologies, methodologies and tools for their development will be indispensable. Evaluation is going to be key, with anyone being able to post an ontology on the web and make it accessible to others. The problem of ontology heterogeneity is an inherent problem of the Semantic Web: ontologies will inevitably overlap and cover similar domains and we will need to figure out how to integrate information associated with them and how to create mappings between ontologies themselves and how to use these mappings for reasoning, query answering, data transformation, and other services. We are probably never going to reach the critical mass of ontologies that is necessary for the Semantic Web truly to take off without being able to learn ontologies from texts and web pages that already exists. Similarly, having a large number of web pages annotated with terms from ontologies is another necessary condition for the realization of the Semantic Web vision. And a bulk of those annotations will have to be created automatically.

5. ONTOLOGY APPLICATIONS

Since there is an immense variety of ontology applications available, we have chosen some fields in which ontology solutions are making the difference, by displaying great performance or by adding new capabilities, not already deployed or with which can ease systems' work. We start by visiting the ontology applications related to texts and, as a consequence, the field of knowledge management.

4.4 ONTOLOGIES AND TEXTS

There is a strong tie between ontologies and text written in natural language. The reason is quite obvious: Ontologies try to capture representation of a domain in a formal language. Humans, however, normally formulate their descriptions of a domain using natural language, often captured in texts. This correspondence between ontologies and text in terms of carriers of intended meaning can be exploited in two directions:

- Ontologies can be employed on text processing applications, and

- Ontologies can be extracted from textual sources [5] — what is known as text-based ontology learning.

Both directions are active fields of research and a number of methods and tools for texts processing and for ontology extraction from texts have been developed. We will briefly discuss these two lines in the following.

Concerning the first direction, ontology-based text processing systems are specially useful to process data related to a specific domain on the Web, once keyword-based search engines, although robust, proved to be an inherently imprecise tool for information search - the output of them to users' queries usually delivers a great deal of irrelevant documents to unsatisfied users. The main problem of this approach is neglecting the context surrounding the pages, thus letting slip many useful pieces of information that could be processed, if systems could count on some form of a priori knowledge about the main subjects referred by the pages.

Basically two types of solutions were proposed to address this problem:

- Endowing IR systems with intelligence, in the form of reasoning capabilities, what gave rise to the intelligent information agents. Many of the successful solutions of this kind rely on some sort of context definition, like the RETSINA agents [53].

- Endowing the Web itself with intelligence, in the form of semantic annotations of pages, what gave rise to the Semantic Web?

It is worth noting that ontologies play a distinctive role on both solutions, since both should be supported by well-defined contexts, which can be represented by ontologies. Focusing on the former type, the alliance of natural language processing (NLP), IR and ontologies seems to outline a winning alternative to process texts using context, and constitutes a clear opportunity for research and development. We are witnessing at least three typical approaches:

- Ontology-based query expansion, where a user query directed to a search engine is enhanced by synonyms or hyperonyms (superclasses)

related to the concept being searched. These concepts are organized in a domain ontology.

Examples of this approach are the ONAIR system, described in the paper of Paz-Trillo et al in this Special Issue. Robin and Ramalho [47] evaluated the performance gain when the linguistic ontology WordNet [36] supplies query expansion terms for a search.

- Semantic information retrieval systems, in which a corpus is previously annotated according to a domain ontology during the indexing process. The retrieval step involves an inference procedure to check the answers, or augment them with implicit information.

The M&M system [41] provides a graphical interface to retrieve semantically pre-annotated documents about molecular biology. Inferences are carried both during indexing and retrieval, allowing users to pose queries by selecting concepts from the ontology.

- Ontology-based information gathering systems, which usually perform more than one task related to text processing, such as retrieval, classification and extraction.

Examples of this approach can be found in the systems WebKB and MASTER-Web. WebKB [7] applies machine learning over a predefined corpus in order to categorise the pages onto a domain ontology, also extracting information from them. MASTER-Web [13] is a multi-agent system which searches, classifies and extracts data from Web pages referring to a specific domain, defined by an ontology. Each agent processes a class of pages, represented by a concept in the domain ontology, and the agents cooperate among themselves to accelerate the search for useful pages.

This type of system can also be employed for the semi-automatic generation of semantic annotations on Web pages. In this context, special syntactic and grammatical patterns can be defined that identify certain types of objects (e.g. People, Places, Events, etc). If these patterns are linked to the corresponding concepts in the ontology, instances of these classes can be identified in documents and annotated with their type.

General semi-automatic annotators are being deployed to assist users on the annotation task (a good list can be found at http://annotation.semanticweb.org/tools/), but there are also information gathering systems with an automatic annotator component. For instance, MUMIS [28] automatically annotates multimedia content according to a domain ontology. It does so by exploiting multiples sources of information about a same event written in multiple languages. Its case study was the processing of video clips' comments about football matches. MUMIS also relies on NLP techniques.

A technology that underlies both information gathering systems, semi-automatic annotation and our second direction to be exploited on ontologies and texts, ontology learning, is pattern-based information extraction. This branch of text processing tries to identify specific pieces of information in texts, either by words or concepts surrounding the information or by finding special types of words based on the grammatical structure of the sentences in which the information appear.

In ontology learning, pattern-based techniques are used to identify words that are useful input for the learning method because they are likely to represent concepts or relations. Special syntactic patterns have been developed for instance for identifying ontological relations like hyponomy [22] or mereonymy [4].

A wide range of methods have been developed for text-based ontology learning. These methods use results from machine learning and natural language processing. Text clustering can be used to group texts that are related content-wise. Dominant terms occurring in a cluster can then be extracted and used to create concepts of a new ontology. Using hierarchical clustering techniques, this method also generates an initial concept hierarchy [12]. Other approaches use formal concept analysis to construct concept hierarchies based on common features of extracted concepts [6]. Learning relations between concepts requires the use of different techniques from machine learning. Existing approaches use association rule learning [31] or Inductive Logic Programming [39] for this purpose.

5.2 KNOWLEDGE MANAGEMENT

Knowledge management is strongly intended to avoid a famous quote from Kevin Abley: "An organisation's knowledge walks out of the door every night – and it might never come back" [10]. The knowledge generated inside an enterprise, in the form of useful information, expertise, best-practices, human resources capabilities and other kinds of knowledge, can be considered its dearest asset for problem solving. And ontologies consist in the adequate technology that can encompass the precise, unambiguous definitions, relations and restrictions that will represent the context of employees' daily routine, and make them register their experiences and findings, collaborate and attain collective intelligence. Furthermore, requirements for KMSs match quite a lot with ontology benefits, such as supporting integration of already existing systems/databases and the ability of coming up with implicit information. An example of the latter is discovering competitors of an enterprise's product, which is actually a deduction from the fact that

competitors are other enterprises which sell products of the same kind of the ones supplied by the referred enterprise [51].

Text processing (TP), as mentioned in the last subsection, is indeed useful, and together with data mining, helps finding and extracting relevant pieces of information from bulks of text and raw data. Although vendors still sell the accomplishment of these tasks as knowledge management, with the appearance of ontologies, text processing has become only the first step of the process. In fact, from an organisational perspective, there are lots of gaps to be filled and problems to reuse these pieces of information only with TP. For instance, there is the *problem* of *positioning* the extracted information. To tackle this problem, an enterprise must rely on knowledge management systems (KMS) that, on its turn, ought to furnish a structure in which useful information can be easily stored in its correct granularity, position and context, thus being transformed into valuable, durable information [44]. This structure must also stress how distinct data relates, and for what sorts of tasks performed by the enterprise, can it be reused or adapted.

An example of solution that fulfils the requirements above is the KnowMore architecture for organisational memories' information systems [1]. It comprehends three layers: (a) the object level, where lies the information sources, like databases, documents, messages, etc, (b) the description level, mainly composed by ontologies to describe and connect the knowledge about the enterprise, its domain of action and the types of information available, together with systems to process object-level data, and (c) the application level, where knowledge-intensive tasks are carried out, profiting from the ontologies of lower levels.

As a consequence of ontologies suitability to these jobs, markets for ontology-based knowledge management systems (OKMSs) [32] are blooming. Big companies like Schlumberger, Daimler Chrysler and BTelecom are investing heavily in knowledge management/ organizational memories solutions based on ontologies as competence differentials, either creating their own KM/ OM departments, providing fundings for European research projects or counting on smaller software development enterprises like Intraspect, Tacit Knowledge and OntoPrise.

6. ARTICLES' OVERVIEW

This Special Issue received 16 submissions, which were peer-reviewed by a high-quality team of ontology researchers, who selected the very best ones. The articles published in this issue have undergone a plenty of changes suggested by these referees, in order to achieve their best quality. We will describe these articles briefly in the following. The first three articles contribute with ontology applications, whilst the others bring ontology descriptions fro the domains of ecology and multi-agent reputation.

The paper entitled "An Information Retrieval application using Ontologies", by Christian Paz-Trillo, Renata Wassermann and Paula Braga, presents a typical ontology-based query expansion IR system, as described in subsection 5.1. The system works over a collection of video clips about contemporary art, a domain described by an ontology written by the authors in OWL. An increase of performance in terms of documents' relevance with the query expansion is reported, according to the results depicted.

The paper "On the Design of Ontology-driven Workflow Flexibilization Mechanisms", written by Tatiana Vieira, Marco A. Casanova and Luis G. Ferrão, addresses the rigidity of workflows by offering them flexibilization mechanisms, which are quite useful in the absence of incomplete or negative information. The authors take advantage of an Workflow Ontology to support two flexibilization mechanisms: (a) presuppositions, with the function of proceeding to workflow execution when in the presence of incomplete information, and (b) a mechanism to determine optional workflow steps, when negative information is met.

The paper "The Use of an Enterprise Ontology to Support Knowledge Management in Software Development Environments", by Karina Vilela, Gleison Santos, Lílian Schnaider, Ana Regina Rocha and Guilherme Travassos, describes what they call "Enterprise-Oriented Software Development Environments" (EOSDE), which stands for Software Development Environments aided by knowledge management components. A central component in their approach is the Enterprise ontology, which is the unifying tie among three EOSDE tools, which are also described.

The paper "EcoLingua: a Fornal Ontology for data in Ecology", by Virgínia Brilhante, reports on a thoroughly detailed Ecology ontology, which reused plenty of quantitative concepts, like dimensions and physical quantities, from the EngMath family of ontologies. The article also sketches an application of the ontology, the synthesis of ecological models.

The last paper, entitled "Using a Functional Ontology of Reputation to Interoperate Different Agent Reputation Models", by Sara Casare and Jaime Sichman introduces an ontology to be used in multi-agent systems, about the different models and respective terminologies employed in each of these models. The authors claim that this ontology could play an important interoperability role in agents' interactions, even when these agents uses different reputation models.

Last but not least, we appreciate the support provided by the former JBCS editor-in-chief, Carlos Masiero, as well as from the current editor, Jaelson Castro. In addition, we specially would like to thank the referees of this Special Issue, our colleagues, Andreas Abecker, Troels Andreasen, Klaus-Dieter Althoff, Richard Benjamins, Djamal Benslimane, Guilherme Bittencourt, Paolo Bouquet, Virginia Brilhante, Oscar Corcho, Marc Ehrig, Giancarlo Guizzardi, Nicola Henze, Martin Hepp, Yannis Kalfoglou, Michel Klein, Eduardo Mena, Boris Motik, Leo Obrst, Adam Pease, Sofia Pinto, Jorge Santos, Sergio Tessaris, Ingo Timm, André Valente, Petko Valtchev, Ubbo Visser, Renata Wassermann and Martin Wolpers. These ontology researchers accomplished a serious, rigorous, and superb reviewing work, without which the making of this Special would not be possible. Jerome Euzenat, Dieter Fensel and Martin Wolpers also helped in finding and assigning papers to the right researchers according to their expertise – we are also indebted to them.

REFERENCES

[1] Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O. & Sintek, M. 1998. "Toward a technology for organizational memories", IEEE Intelligent Systems 13(3), 40-48.

[2] Arens, Y., Hsu, C.-N., and Knoblock, C. A. 1996. Query processing in the SIMS information mediator. In Advanced Planning Technology. AAAI Press, Menlo Park, CA.

[3] Arpírez, J. C., Corcho, O., Fernández-López, M., Gómez-Pérez, A. 2003. WebODE in a Nutshell. Artificial Intelligence Magazine. Volume 24, no. 3, Fall, American Association for Artificial Intelligence Press. 37-47. Menlo Park, CA, USA

[4] Berland, M. and Charniak, E. 1999. Finding Parts in Very Large Corpora. In Proceedings of the 37th Annual Meeting of the ACL.

[5] Buitelaar, P., Cimiano, P., Magnini, B. 2005. Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press.

[6] Cimiano, P., Hotho, A., Staab, S. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Anaylsis Journal of Artificial Intelligence Research (JAIR) 24: 305-339. August 2005.

[7] Craven, M., McCallum, A., DiPasquo, D., Mitchell, T., Freitag, D., Nigam, K., Slattery, S. 1999. Learning to Extract Symbolic Knowledge from the World Wide Web. Technical Report CMU-CS-98-122. School of Computer Science. Carnegie Mellon University, USA

[8] Decker, S., Erdmann, M., Fensel, D., and Studer, R. 1999. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editor, Semantic Issues in Multimedia Systems. Proceedings of DS-8, pages 351–369. Kluwer, Boston, MA.

[9] DeMarco, T. 1979. Structured Analysis and System Specification. Prentice-Hall.

[10] Dzbor, M. 1999. KNOWWEB research project and KNOWWEB toolkitkmi.open.ac.uk/people/dzbor/public/knowweb1/KnowWebPaper.doc

[11] Ehrig, M. and Staab, S. 2004. QOM - Quick Ontology Mapping. In 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan.

[12] Faure, D. and Nedellec, C. (1998). ASIUM:learning subcategorization frames and restrictions of selection. In Kodratoff, Y., editor, Proceedings of Workshop on Text Mining, 10th European Conference on Machine Learning (ECML 98).

[13] Freitas, F., Bittencourt, G 2003 An Ontology-based Architecture for Cooperative Information Agents. Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI '2003), Acapulco, Mexico.

[14] Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. 2003. Sweetening wordnet with DOLCE. AI Magazine, 24(3):13–24.

[15] Garcia-Molina, H., Papakonstantinou, Y., Quass, D.,Rajaraman, A., Sagiv, Y., Ullman, J., and Widom, J. 1995. The TSIMMIS approach to mediation: Data models and languages. In Next Generation Information Technologies and Systems (NGITS-95), Naharia, Israel.

[16] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. 2004. Semantic matching. In 1st European semantic web symposium (ESWS'04), pages 61–75, Heraklion, Greece.

[17] Gómez-Pérez, A., Fernández-López, M., Corcho, O. 2004. Ontological Engineering, Springer Verlag, Germany.

[18] Gruber, T. 1995. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies, 43(5/6): 907-928

[19] Gruber, T. 1996. Ontolingua: A mechanism to support portable ontologies. Technical report KSL-91-66. Stanford University, Knowledge Systems Laboratory. USA.

[20] Grüninger, M. and Kopena, J. 2005. Semantic Integration through Invariants. AI Magazine 26(1): 11-20.

[21] Guarino, N. 1996. Understanding, Building, And Using Ontologies. Proceedings of 10th Knowledge Acquisition for Knowledge-Based Systems Worshkop. Banff, Canada.

[22] Guarino, N. Welty, C. 2002. Evaluating Ontological decisions with OntoClean. Comm.of The ACM(2):61-65

[23] Handschuh, S. Staab, S. (eds.). 2003. Annotation for the Semantic Web. IOS Press, Netherlands.

[24] Hearst, M. 1992. Automatic Acquisition of Hyponyms in Large Text Corpora. Proceedings of the Fourteenth International Conference on Computational Linguistics.

[25] Heflin, J., Hendler, J., and Luke, S. 1999. SHOE: A knowledge representation language for internet

applications. Technical Report CS-TR-4078, Institute for Advanced Computer Studies, University of Maryland.

[26] Hovy, E. 1998. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In The First International Conference on Language Resources and Evaluation (LREC), 535–542, Granada, Spain.

[27] Kalfoglou, Y. and Schorlemmer, M. 2003. Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1):1–31.

[28] Kuper, J., Saggion, J., Cunningham, H. Declerck, T., de Jong, T., Reidsma, D., Wilks, Y. Wittenburg, P. 2003. Intelligent Multimedia Indexing and Retrieval through Multi-source Information Extraction and Merging_ Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI '2003), Acapulco, Mexico.

[29] Lenat, D. 1990. Towards Programs with Common Sense. Communications of ACM. 33(8):30-49. USA

[30] Levy, A., Rajaraman, A., and Ordille, J. J. 1996. Querying heterogeneous information sources using source descriptions. In Proceedings of the 22nd International Conference on Very Large Data Bases, VLDB 1996.

[31] Maedche, A. and Staab., S. 2000. Discovering Conceptual Relations from Text. In Horn, W., editor, Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI 2000), pages 321 – 325, Berlin, Germany. IOS Press.

[32] Maedche, A., Motik, B., Stojanovic, L., Studer, R., Volz, R. 2003. Ontologies for Enterprise Knowledge Management.*IEEE Intelligent Systems* 18 (2): 26-33.

[33] McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S. 2000. An environment for merging and testing large ontologies. In Cohn, A. G., Giunchiglia, F., and Selman, B., editors, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000). Morgan Kaufmann Publishers, San Francisco, CA.

[34] Melnik, S., Garcia-Molina, H., and Rahm, E. 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In 18th International Conference on Data Engineering (ICDE-2002), San Jose, California. IEEE Computing Society.

[35] Mena, E. and Illarramendi, A. 2001.Ontology-Based Query Processing for Global Information Systems. Kluwer.

[36] Miller, G. 1995. WordNet: A Lexical Database for English. Communications of the ACM. 38(11):39-41. USA.

[37] Mitra, P., Wiederhold, G., and Kersten, M. 2000. A graph-oriented model for articulation of ontology

interdependencies. In Proceedings Conference on Extending Database Technology 2000 (EDBT'2000), Konstanz, Germany.

[38] Neches, R., Fikes, R., Finin, T., Gruber, T.R.,Patil, R., Senator, T., Swartout, W. 1991. Enabling Technology for Knowledge Sharing. Artificial Intelligence Magazine. 12(3), American Association for Artificial Intelligence Press. 36-56. Menlo Park, CA, USA

[39] Nédellec, C.. 1999. Corpus-based learning of semantic relations by the ILP system, Asium. In James Cussens, editor, Proceedings of the 1st Workshop on Learning Language in Logic, pages 28—39, Bled, Slovenia.

[40] Niles, I. and Pease, A. 2001. Towards a standard upper ontology. In The 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine.

[41] Noy, N. 1997. Knowledge Representation for Intelligent Information Retrieval in Experimental Sciences. PhD Thesis. Computer Science Faculty, Northeastern University of Boston, MA, USA.

[42] Noy, N. F. and Musen, M. A. 2003. The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59(6):983–1024.

[43] Noy, N., Fergerson, R., Musen., M. 2000. The knowledge model of Protege-2000: Combining interoperability and flexibility. 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France

[44] O'Leary, D. 1998. Knowledge Management Systems: Converting and Connecting. IEEE Intelligent Systems 13(3), 30-33

[45] Ontoprise produkts. 2003.www.ontoprise.de/produkts

[45] Rapoza, J. 2000. DAML could take search to a new level. PC Week, February 7,2000. USA.

[47] Robin, J., Ramalho, F. 2001 Empirically evaluating WordNet-based query expansion in a web search engine setting. Proceedings of IR'2001, September 19-21, Oulu, Finland,.

[48] Shortliffe, E. 1976. Computer-based Medical Consultations: MYCIN. Elsevier, NY.

[49] Sowa, J. 1999. Knowledge representation: logical, philosophical and computational foundations. Brooks Cole Pub. Co. Pacific Grove, USA

[50] Studer, R., Benjamins, R., Fensel, D. 1998. Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering. 25(1998):161-197.

[51] Studer, R., Volz, R. 2003. Ontologies:Representation, Engineering, Learning & Applications, Tutorial slides. 3rd IFIP Conference on E-Commerce, E-Business and E-Government (I3E´03) September 21st. Sao Paulo, Brazil

[52] Swartout, B., Patil, R., Knight, K., and Russ, T. 1996. Toward distributed use of large-scale. ontologies. Proceedings of 10th Knowledge Acquisition for Knowledge-Based Systems Worskhop. Banff, Canada.

[53] Sycara, K. 1999. In-context information management through adaptative collaboration of intelligent agents. In: Klusch, M. (sd.) Intelligent nformation Agents, Springer Verlag, Germany

[54] Uschold, M. and Gruniger, M. 1996. Ontologies: Principles, methods and applications. Knowledge Engineering Review, 11(2):93–155.