Ontologies – in Support of Web Intelligent Agents' Development

Senior Lecturer Vasile - Daniel Păvăloaia, PhD
Al .I. Cuza University of Iasi, Romania
danpav@uaic.ro.

Abstract: The area of agent-based systems is, nowadays, quite popular. At this point, there is a specialized technology required for building software agents that should be communicative, capable, autonomous and adaptive – the key characteristics required to help make the internet activity more successful. The limiting factors in building such systems are being overcome, and new approaches are emerging from information technology research laboratories around the world. The use of ontology has proven to be essential elements in many applications and thus, they have been successfully applied in agent systems technology, knowledge management systems, and e-commerce platforms.

Keywords: Knowledge Acquisition Systems; Artificial intelligence; Knowledge management; Intelligent agents

JEL Classification: M15

1 Introduction

From an Artificial Intelligence approach it is stated that ontology stand for formal models of a shared understanding within a domain. Thus, as the Knowledge Management has its roots in Artificial Intelligence, ontology are considered to be a key technology for Knowledge Management especially due to their aim of bringing a consensus in the way a particular area of expertise is described. This consensus extends not only to terminology, but also to the way concepts and objects may be organized and structured within the domain. Due to the formality[1] of ontology they also have the advantage to be modeled through information systems.

The term ontology[2], comes from the Greek language and stands for: *ὄν - of being* and *λογία(logy) - science, study, theory*.

[1] http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2003_ontohandbook_ens.pdf, pag.1, 11/10/2006.

[2] http://en.wikipedia.org/wiki/Ontology, 10/10/2006

From a philosophy perspective, ontology is a branch of metaphysics, studying the nature of being or existence and its purpose is discovering the entities and their categories that are available.

From an information systems point of view, ontology seeks to express a complete and rigorous conceptual scheme of a certain domain, a hierarchic structure that should contain all the entities, the relationships between them and the rules of the domain.

2 Representative Definitions of Ontology

Ontology can be a computerized model of a certain part of a domain. Frequently, it can be illustrated as a semantic network: a chart whose nodes are concepts that belong to individual object and the arches stand for relationships and concepts associations. The above described network is featuring the following elements: properties, attributes, constraints, functions and rules and they dominate the behavior of concepts.

From a formal perspective, the ontology is a convention of a conceptualization that includes work frames designated for modeling purpose of the field of activity. It also contains conventions that lead to representation of the theory from a domain.

The specialized literature associates definitions of ontology with the name of entities (classes, relationships, functions and other objects) with text information that can be read out by individuals. In the above case, the ontology describes what the name represents, and formal axioms that contain the name interpretation and use.

Thus, ontology are often equated with taxonomic hierarchies of classes, but class definitions, and the subsumption relation, but ontology are not limited to these forms. Ontology is also not limited to conservative definitions, that is, definitions in the traditional logic sense that only introduce terminology and do not add any knowledge about the world.

Ontology helps organizations by allocating a base for domain standard descriptions. Such descriptions can be markup or metadata and they are expressed in terms of the ontology that the members of the domain are committed to.

3 Ontology Languages – A Selective Approach

Ontology has proven to be essential elements in many applications and they have been applied in agent systems technology, knowledge management systems, and e-commerce platforms. Ontology can also imitate natural language, put together information in a wise manner, provide semantic-based access to the Internet, and extract information from texts in addition to being used in many other applications to explicitly declare the knowledge embedded in them.

Web Ontology Language - OWL is a semantic markup language and is developed[1] to be used by applications in which is required the process of the content of information instead of presenting information to human users. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S). It provides additional vocabulary and also a formal semantics frame. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

Another ontology language, The Ontology Inference Layer – OIL is another offer for a web-based representation and inference layer for ontology and it combines the widely used primitives modeling from frame-based languages with the formal semantics and reasoning services provided by description logics[2]. It is quite similar with RDF Schema (RDFS) and contains a precise semantics for describing term meanings. Therefore, it is also useful when describing implied information.

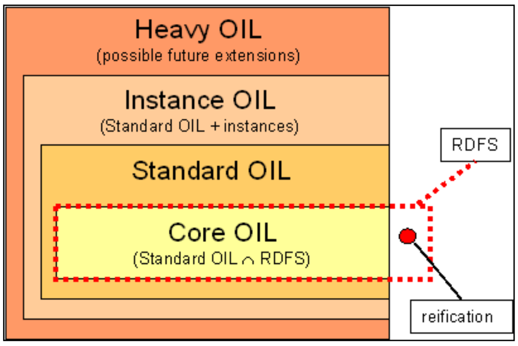

Figure 1. The relation between the layered OIL and RDFS

Source: http://www.ontoknowledge.org/oil

[1] http://www.w3.org/TR/owl-features/ 10/10/2006.
[2] http://www.ontoknowledge.org/oil/ 12/10/2006.

The OIL language has a layered content for a standard ontology language. Each additional layer contributes to the functionality and complexity of the previous layer. In this manner, the agents (humans or machines) who can only process a lower layer can still partially comprehend ontology that are expressed in any of the higher layers. The figure 1 reveals the relation between the OIL dialects and RDFS.

Semantic web is a vision of future web where the information is giving and explicit meaning for easy processing and integrating. Semantic web is relying on the ability of XML standard to define specific XML tag schemes as well as on flexibility of RDF standard in representing data.

4 The Characteristics of an Intelligent Web Agent

On one hand, such an agent should be able to communicate, using a natural language, with the user. Therefore, a common language needs to be established. On the other hand, an agent should be able to act rather than suggest certain steps that are to be followed. An advisor can suggest someone where to go in order to book a plane ticket, a hotel room or a car but when it comes to act, one have to do it on its own. Therefore, an agent should do all the work for us, act in our behalf and take the best decisions. In order for the tasks to be achieved, the agent should be able to collect the right information, search according to the requests, choose the best offer and contact us in order to suggest its findings.

A reliable web agent should have the same abilities with a human agent, namely: should be communicative, able to understand the aims and constraints, be autonomous, adaptable, able to learn from previous experience and fulfill the user preferences.

It is said that a search engine contains agents that runs the search. Let's assume that someone is looking for an article that belongs to the Physics domain. The agent, that runs the search, should be familiar with some Physics concepts instead of just searching for certain keywords. An answer to the above questions would be the use of ontology, meaning that a piece of knowledge should be formally defined.

The most used characteristic of ontology, when it comes to building an agent, involves the use of a structural component. In this case, ontology would consist in the development of taxonomy between object classes and subclasses, accompanied by definitions and descriptions of the relationships between such objects. Besides, ontology contains inference rules, explicit rules that refer to certain objects or structural inferences that are offered by the system. Examples:

IF X is a car

THEN X has 4 wheels

or IF the tire is part of the wheel

THEN the tire is part of the car

If ontology could be understood by a machine, then a computer could handle the terms that the ontology uses, terms that have a certain meaning for the human users which can comprehend such information. A computer cannot understand certain (type of) information, in its profound meaning, but handles terms that the human user doesn't always comprehend.

Therefore, the establishment of a common channel is a must and it allows them (computer and user) to understand each other and furthermore, permits the development of intelligent agents that can handle the human needs, preferences and constraints.

5 The Components of an Intelligent Agent Information Chain

The components of an intelligent agent information chain can explain how each link (component) of the chain can lead to information that allows the existence of the next link. In figure 2 there are presented the components of web intelligent agent information.

5.1 Ontology Development Tools for Intelligent Agents

The main aim of ontology development tools for intelligent agents is to guarantee a low cost level at a good quality rate. Similar to human need, knowledge and capabilities, the ontology evolve and encounter changes over time. Thus, cutting acquisition and maintenance costs is a very important task.

Protégé and WebOnto are two examples of ontology editors applied in the development of knowledge acquisition systems.

Protégé is a free, open source ontology editor and knowledge-base framework. Protégé is a platform that supports two main modeling ontology manners that use frames and OWL editors. Furthermore, the Protégé ontology can be exported into a variety of formats including RDF(S), OWL, and XML Schema. Protégé is based on Java technology, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

The Protégé Programming Development Kit (PDK)[1] contains a set of documents designed to describe and illustrate the manner in which a plug-in extension for Protégé should be developed and installed. Protégé also has an OWL API that extends the core API to provide access to OWL ontology and can be used directly by external applications to access Protégé knowledge bases and make use of Protégé forms without running the Protégé application.

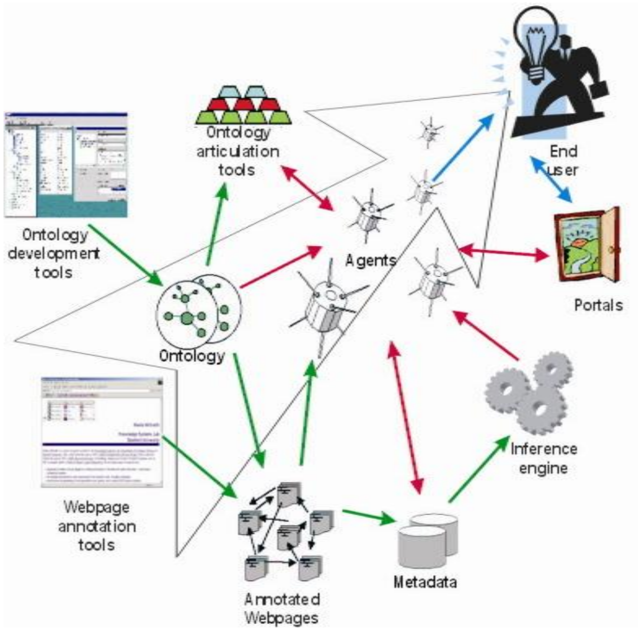

Figure 2 The components of an intelligent agent information chain

Source: www.aifb.uni-karlsruhe.de/WBS/Publ/2000/ECDL_sdeetal_2000.ps

The main aim in designing WebOnto was for the purpose of supporting the collaborative browsing, creating and editing of ontology without having interface problems related to standard HTML-based editors, such as limited support for direct manipulation interfaces, inability to handle asynchronous communication and limited support for graphical interfaces. In order to avoid the above problems WebOnto was implemented as a Java Applet[2], and includes both graphical user interface and fine grained inspector windows. WebOnto also provides a fruitful set

[1] http://protege.stanford.edu/
[2] http://kmi.open.ac.uk/projects/webonto/, 10/10/2006.

of options to customize the presentation of information, making it easier to navigate through large ontology. WebOnto is available as a public service and can be launched from the web.

5.2 Webpage Annotation Tools

The annotation process requires specialized tools that assist the user during the development process of webpage annotation projects. By annotation there is understood any comment, note, explanation, or other type of external remark that can be attached to any whole or selected part of a Web document, without actually needing to change the document. When the user gets the document it can also load the annotations attached to it from a selected annotation server or several servers and see what the peer group thinks.

There are several specialized tools available[1] and, among them, we will refer to only few:

- SHOE Knowledge Annotator is software based on Java platform and allows users to mark-up web pages with SHOE knowledge without having to worry about the HTML codes[2].
- Annotea, a LEAD[3] project enhancing the W3C collaboration environment with shared annotations, is open source, part of the Semantic Web efforts and it uses and helps to advance W3C standards when possible. As a result, it uses an RDF based annotation schema for describing annotations as metadata and XPointer for locating the annotations in the annotated document. The annotations are stored on annotation servers under metadata forms and presented to the user by a client capable of understanding this metadata and capable of interacting with an annotation server that has the HTTP service protocol[4].
- Ontopad is an extension of a Java-based HTML editor, which allows normal browsing and editing of the HTML page, and supports the annotation of the HTML-page with ontology-based metadata. The annotator can select a portion of the text from a webpage and choose to add a semantic annotation, which is inserted into the HTML source. However, for significant annotation tasks a

[1] http://annotation.semanticweb.org/tools/, 09/10/2006
[2] http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html, 08/11/2006
[3] LEAD - Live Early Adoption and Demonstration
[4] http://www.w3.org/DesignIssues/Architecture.html#Collaboration, 08/11/2006

basic annotation tool is not sufficient. It still takes a longtime to annotate large pages, although a significant improvement was reported when compared to the manual task. Thus, a practical tool should also exploit information extraction techniques for semi-automatic metadata creation. The precision of linguistic processing technology is far from perfect. Also, reasonably exact automatic annotation is not yet possible.

5.3. Ontology Articulation/Interconnecting Tools

Ontology Articulation consists in aligning two ontologies O1 and O2 by means of a pair of ontology mappings from intermediate source ontology O3. The resulting ontology O3, together with its mappings, is called the articulation of two ontologies. Articulation allows defining a manner in which the fusion or merging of ontology has to be fulfilled.

For the purpose of solving a task that involves consulting multiple information sources that have separate ontology, an un-automated agent requires to link the semantic gap between the several ontology found on the web. In the Scalable Knowledge Composition Project1 the authors developed tools that assist an expert with the process of defining rules and bridge ontology from different sources that have different terminology and semantic relationships. The obtained rules define new articulation ontology, which help the application and translate terms.

The major advantage in using the SKC approach is that not all of the terms in the source ontology have to be aligned in order to be made globally consistent. Aligning completely just two ontologies requires a major effort for a practical application, as well as ongoing maintenance.

Another important characteristic of articulation is scalability. Since there are hundreds of domains just in the logistics area, and different applications need to use them in various combinations, the global-consistency approach would require that all domains which interact must be made consistent. No single application can take that responsibility, and it is unlikely that even any government can mandate national semantic consistency.

[1] Mitra, P., Wiederhold, G., and Kersten, M., A Graph-Oriented Model for Articulation of Ontology Interdependencies, Germany, 2000.

5.4 Agent's Inference Engine

The use of inference engines can reduce metadata creation and maintenance cost[1]. Every single assertion of metadata must explicitly lead to a large metadata creation and maintenance overhead. Therefore tools and techniques are necessary to help reducing the amount of explicitly stated metadata by inferring further implicit metadata[2.] Implicit metadata can be derived from already known metadata by using general background knowledge, as stated by the ontology.

Over the Web, metadata is distributed and the added value can be generated by combining metadata from several metadata offerings: for instance, for the travel area, metadata can state that between Iasi and Bucharest a flight is available. Another statement, issued by another server, states that a flight between Bucharest and Amsterdam is available. Combining the above two pieces of information an agent inference system can infer that a connection between Iasi and Amsterdam is available although this information it not made explicit by any servers (see figure 3).

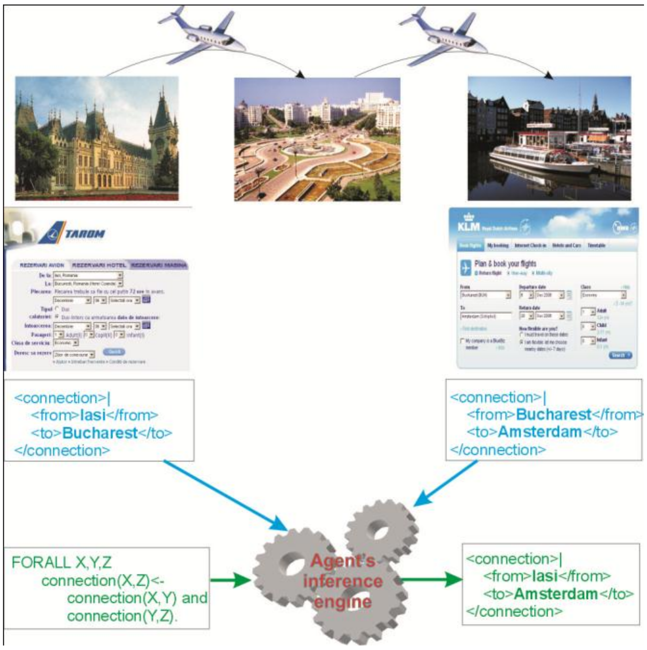

Figure 3 Agent inference system

[1] www.aifb.uni-karlsruhe.de/WBS/Publ/2000/ECDL_sdeetal_2000.ps.
[2] http://www.w3.org/TandS/QL/QL98/pp.html.

Adapted after: www.aifb.uni-karlsruhe.de/WBS/Publ/2000/ECDL_sdeetal_2000.ps

Such simple rules as those showed in Figure 3 cannot be considered by commercial databases because of their inability to perform recursion (they can generate infinite looping).

6 Conclusions

Within the projects mentioned in the current material, there were presented real possibilities to annotate web page throughout ontology based metadata. The architectures based on intelligent agents are being based on the communication between agents and less on ontology development and delivering. The informational chain model offers a focused infrastructure based on an automatic web infrastructure throughout agents. The model is based on web semantic methodology and assumes that the web collection of data should become less a collection of HTML documents and more of a source of formalized knowledge where the software agents should be in charge.

7 Acknowledgement

„This work was supported by the project "Post-Doctoral Studies in Economics: training program for elite researchers - SPODE" co-funded from the European Social Fund through the Development of Human Resources Operational Programme 2007-2013, contract no. POSDRU/89/1.5/S/61755)".

8 References

Davies, J., Fensel, D., Harmelen, F., editors (2003). *Towards the Semantic Web: Ontology-Driven Knowledge Management*. John Wiley & Sons.

Decker, S., Erdmann, M., and others (1999). *Ontology Based Access to Distributed and Semi-Structured Unformation*. Kluwer Press.

Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, Formal *Ontology in Conceptual Analysis and Knowledge Representation, Deventer*. The Netherlands: Kluwer Academic Publishers.

Mika, P. (2002). Integrating Ontology Storage and Ontology-based Applications Through Client-side Query and Transformations. *Proceedings of Evaluation of Ontology-based Tools (EON2002) workshop at EKAW2002,* Siguenza, Spain.

Mitra, P., Wiederhold, G., and Kersten, M. (2000). A Graph-Oriented Model for Articulation of Ontology Interdependencies. *Proceedings Conference on Extending Database Technology 2000 (EDBT'2000)*, Konstanz, Germany.

The Ariadne Project at the Information Sciences Institute – www.isi.edu/ariadne

The European Community Agentlink project – http://www.agentlink.org/

The French "@gency" Frontpage – http://hoegaarden.iutc3.unicaen.fr/cgi-bin/pywiki

The MIT Media Laboratory Agents Research group – http://agents.www.media.mit.edu/groups/agents/

The National Research Council of Canada Agent resource list – http://ai.iit.nrc.ca/subjects/Agents.html

The On-To-Knowledge project (EU-IST-1999-10132). http://www.ontoknowledge.org.

The UMBC Agent Web – http://www.cs.umbc.edu/agents/

The University of Washington "Softbots" Project – http://www.cs.washington.edu/research/projects/softbots/www/softbots.html