State of the Art: Ontologies

Javier Gramajo López

{jgramajo}@saesap.usac.edu.gt
Engineering Faculty, Computer Since and Systems Department
Universidad San Carlos de Guatemala
April 2005

Abstract. This document is a compilation related with Ontologies that will used as a reference document in the course of Artificial Intelligence.

Keywords: Ontologies, Knowledge engineering, discovery, representation, acquisition, reusing, behavior.

1 Introduction

The World Wide Web has shift the way people get information and order services. However, with the rapid development of the World Wide Web, the problem of searching and presenting information has also turned out: finding the right piece of information is often a problem since searches are imprecise and often produce matches to many thousands of pages. Users face the task of reading the documents retrieved in order to extract the information desired.

The explicit representation of the semantics underlying data, programs, pages, and other web resources, will enable a knowledge-based web that provides a qualitatively new level of service. Automated services will improve in their capacity to assist humans in achieving their goals by "understanding" more of the content on the web and thus provide more accurate filtering, categorization, and search of information sources.

Ontologies provide a shared and common understanding of a domain that can be communicated between people and heterogeneous application systems. An ontology provides an explicit conceptualization that describes the semantics of the data. They have a similar function as a database schema. Ontologies are needed to help the agents and analysis tools to retrieve information on the web. Ontologies will play a key role in areas such as knowledge management, B2B e-commerce, etc. In [1] an Ontology is described as a key that enables the appropriate technology for *The Semantic Web*.

We plan to use Ontologies as a description of a specific domain, this ontology and their properties are used to find a list of documents that contain information related to the ontology.

Because of this, in our research work we will use Ontologies that were created by an expert who certainly knows the domain structure, relations between the classes and their properties, this ontology will be created using some ontology oriented representation standard, this standard will be selected over the available standards, a parser and an editor will be selected too.

2 What is an Ontology

Ontologies were developed in Artificial Intelligence to facilitate knowledge sharing and reuse. Recently Ontologies have become widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management. An ontology is a formal and explicit specification of a shared conceptualization [2]. Ontologies are developed specially to enable this new type of knowledge representation and they have become a popular research topic in various communities, such as knowledge engineering, natural language processing, co-operative information systems, intelligent information integration, and knowledge management. They provide a shared and common understanding of a domain that can be communicated between people and application systems. Ontologies are an essential backbone technology because they provide an important feature: they interweave formal semantics understandable by a computer with real world semantic understandable to humans. An ontology can provide a standard vocabulary for a problem domain. Typically, an ontology contains a hierarchical description of important concepts in a domain, and describes crucial properties of each concept through an attribute-value mechanism. Further relations between concepts may be described through additional logical sentences. However, an ontology can also contain structures or axioms that define the semantics of the vocabulary terms. These semantics can be used to infer information based on background knowledge of the domain and to integrate data sources from different domains.

3 The Usage of Ontologies

Some areas that develop, construct and make use of Ontologies are: Engineering, Medical, Teaching, and Information System Development. Ontologies can be used during the life cycle of information system development. For example, during the requirement stage we can use ontology to obtain knowledge about the domain, and also we can use it for requirements specification. More importantly, ontology can be used for agreement on the validation of information system requirement between the stakeholders. In addition, Ontology can help in the construction of the conceptual modelling of the discussion domain.

An enterprize Ontology is a collection of terms and definitions. This kind of Ontologies are developed as part of the enterprize project, which is a team effort to provide a structure for enterprize modelling. This Ontology is responsible for acting as a communication intermediary between the different people, including users and developers, across different enterprizes, and between people and implemented computational systems.

Ontology can also be used in various problem-solving tasks. Byung [3] has used his proposed language to analyze various domains. He constructed an Automobile Cooling System Ontology and found that the same Ontology can be used to explain the structure of the domain and to find the cause for a given fault diagnosis.

Ontology concepts can be certainly applied to technology intensive industries and these have to deal with rapidly changing knowledge domains. Ontologies can be developed to describe such domains and applied to serve a number of purposes, such as the use of Ontologies to refine user queries while searching in a knowledge database or to assist the search for information on the internet as we plan to use it in GINY.

Some examples of representative knowledge domains of the Technology intensive industries may include :

- Technology and employment markets domain.
- Environmental and sustainable development domain or.
- Product-oriented supply markets domain.

Ontologies can be used to:

- Classify information related to each knowledge domain.
- Allow users to receive information related to a domain.
- Define the depth of hierarchical level in the classified information that a user is interested in.
- Continuously follow-up and update received information.
- Monitor relative information contributions.

4 Types of Ontologies

Ontologies aim at consensual domain knowledge and their development is often a cooperative process involving different people, possibly at different locations. Depending on their generality level, different types of Ontologies may be identified that fulfill different roles in the process of building a Knowledge Based System (KBS). The ontology types are divided into the following categories [4]:

- *Domain Ontologies*: which capture the knowledge valid for a particular type of domain (e.g. chemical, electronic, medical, mechanic, digital domain).
- *Metadata Ontologies*: like Dublin Core for describing the content of on-line information sources.
- *Generic or common sense Ontologies*: aim at capturing general knowledge about the world, providing basic notions and concepts for things like time, space, state, event etc.
- *Representational Ontologies*: do not refer to any particular domain and provide representational entities without stating what should be represented.
- *Task and method Ontologies*: Task Ontologies provide terms specific for particular tasks, and method Ontologies provide terms specific to particular Problem-Solving Methods (PSMs). Task and method Ontologies provide a reasoning point of view on domain knowledge.
- *Application Ontologies*: that contain all the necessary knowledge for modelling a particular domain (usually a combination of domain and method Ontologies)

5 Standardized Knowledge Representation Frameworks

The immense growth of the World Wide Web in the past decade has showed in an era in which the development of the Semantic Web is the key to support the excess of information available into tangible knowledge. The World Wide Web, where data is mainly used by humans, the new generation will provide data that is also machine processable. Tim Berners-Lee called this new generation at his book "Weaving the Web" the *Semantic Web* [5]. The Semantic Web is not just a simple repository of data. It describes rules and meanings about data, which are defined so that machines can quickly and without errors interpret and process that data. The Semantic Web is knowledge and the Ontologies have appeared to code this kind of knowledge. Figure 1 shows the Semantic Web proposed layers, presented by Berners-Lee at the XML 2000 conference with higher-level languages using the syntax and semantics of lower levels [6].

Fig. 1. Proposed layers of the Semantic Web.

There is a direct interaction between Semantic Web and Ontologies. That is, the one is linked to the other and vice versa. The role of Ontologies for the Semantic Web is so important that it led to the creation of new extension of web standards with ontology oriented language concepts. Examples of these web standards are XML, which has been extended to give XML-Schema and also RDF, which has been extended to give RDF-Schema. Furthermore, Semantic Web and Ontologies will be vital for the development of web applications by providing to the users more efficient search and browsing.

An important requirement for machine-processable information is structure in data. There are several ways to structure data: databases use tables with attributes as structuring mechanism. The main structuring technique that is used on the web is document markup. A markup language is a specification of the markup-tags that may be used, how they may combined, and what their meaning may be. HTML is the most well-known and wide spread language on the web. It provides tags to specify abstract rendering information for text documents. However, HTML only provides rendering information about the content, which is often not sufficient for the advanced automated services that are foreseen on the

Semantic Web. Applications will need specialized markup that specifies the role of parts of the content that are relevant for them.

5.1 XML

The eXtensible Markup Language (XML) [7] is a meta-language that meets the need to define application specific markup tags. XML is a mechanism for representing other languages in a standardized way. XML is a standard language that simultaneously presents content for display on the Internet and describes the content so that other software can understand and use the data. Therefore XML has the potential to be a medium through which any business application can share documents, transactions and workload with any other business application. In other words, XML can become the common language of e-business and knowledge management. XML offers capability important to Knowledge Management Systems (KMS): simplification of indexing. The resource description framework (RDF) specification defines a data model and syntax. The syntax is defined on top of the XML syntax. The data model is defined in terms of resources often identified with Uniform Resource Identifiers (URIs). Some of the resources are "anonymous". The data model is a set of triples, often thought of as a graph. The anonymous resources correspond to blank nodes in the graph.

5.2 RDF and RDF Schema

While XML is a standard mechanism to structure data, the Resource Description Framework (RDF) [8] is a mechanism to tell something about data, i.e., to give meaning to it. As its name indicates, it is not a language but a model for representing data about "things on the web." This type of data about data is called meta-data. The "things" are resources in RDF vocabulary. Resources are all things that have an identifier on the web, ranging from specific, identifiable parts of a document to things like persons or companies. RDF Schema is a simple type system for RDF. It provides a mechanism to define domain-specific properties and classes of resources to which you can apply those properties. RDF and RDF Schema provides a simple knowledge representation mechanism for web resources. RDF is the emerging standard proposed by the W3C for the representation and exchange of metadata on the Semantic Web. RDF Schema is the standard dedicated to the representation of ontological knowledge used in RDF statements. The basis of RDFs strength as a knowledge-management tool is that it allows user to organize, interrelate, classify, and annotate this knowledge, thereby increasing the aggregate value of the stored data. RDF has a reputation for complexity that is contradicted by the simplicity of adding RDF support to XML-based applications. RDF Schema is quite simple compared to full-fledged knowledge representation languages. It only has a few modelling primitives, and does not have exact, logical, semantics. To be able to specify the meaning of data more precisely, richer languages are necessary.

5.3 Ontology Interchange Layer (OIL)

OIL [8] is such an enhanced language. OIL is the result of an initiative to integrate intuitive modelling primitives, web-languages, and formal semantics into one language. OIL's onion model offers languages of varying complexity; this allows applications to select the degree of complexity they require.

OIL is a frame-based standard proposal for ontology interchange language that might include the necessary core language primitives and provide the developers with the possibility to create their own extensions of the language.

OIL is a joint standard for specifying and exchanging Ontologies and based on OKBC (Open Knowledge Base Connectivity, is an Application Program Interface (API) for accessing frame-based knowledge representation systems), XOL (is based on XML and the modelling primitives, and its semantics is based on OKBC-Lite) and RDF (Resource Description Framework is an infrastructure that enables the encoding, exchange and reuse of structured metadata).

5.4 DAML + OIL

DAML+OIL is currently used by the W3C Web-Ontology (WebOnt) [9] Working Group as a starting point for a W3C Ontology Web Language (OWL). W3C RDFS provides primitive classification and simple rules for this, but DAML+OIL goes much further. Users might see in this some of the flavor of business rules, which are known in software development circles as the programmatic expression of mandates for the way data must be processed. In fact, one way to look at DAML+OIL is as the business rules for the Semantic Web, yet it is much more flexible than most business-rules-languages in common use.

DAML+OIL reflects a broad European and (US) American consensus on modelling primitives for the Semantic Web and will be the departure point for standardization by the W3C. Both OIL and DAML+OIL are defined as an extension to RDFS, which makes them to a large extent compatible with plain RDFS. Their main additions to RDFS are formal semantics, based on a Description Logic, and more advanced modelling primitives, such as boolean expressions and some axioms.

Fig. 2. Knowledge Representation Frameworks

Figure 2 shows a general view with above presented Knowledge Representation Frameworks and their relation.

5.5 The Simple HTML Ontology Extension (SHOE)

SHOE [10] has been developed by the Parallel Understanding Systems Ground in the Department of Computer Science at the University of Maryland. SHOE is intended to provide user agents with easy access to machine readable semantic knowledge on the web. It does this by adding to HTML a simple knowledge-representation language. SHOE tags are divided into two categories. Firstly, there are tags for constructing Ontologies. SHOE's Ontologies are sets of rules, which define what kinds of assertions SHOE documents can make and what these assertions mean. Secondly, there are tags for annotating SHOE documents to subscribe to one or more Ontologies, declare data entities, and make assertions about those entities under the rules prescribed by the Ontologies. SHOE's semantics is intentionally very simple.

5.6 The W3C Web Ontology (OWL)

OWL [11] is applying lessons learned so far with DAML+OIL to develop a semantic markup language for publishing and sharing Internet-based

Ontologies. OWL also builds on RDF and facilitates greater machine-readability of web content by providing additional vocabulary for term descriptions. The W3C has released draft documents that describe the abstract syntax for OWL as well as OWL Lite, a subset of the full language that attempts to capture many of the commonly used features of OWL and DAML+OIL.

6 Semantic Web Implementation Tools: Ontology Editors

Building Ontologies manually is a time-consuming task. It is very difficult and unmanageable to manually derive Ontologies from data. Effective and efficient work with the Semantic Web must be supported by advanced tools enabling the full power of this technology.

Ontology editors help human knowledge engineers to build Ontologies. Ontology editors support the definition of concept hierarchies, the definition attributes for concepts, and the definition of axioms and constraints.

6.1 Protégé-2000

Protégé-2000 [12] is a Java based ontology and knowledge development tool that has been developed at Stanford University. It allows the creation of several custom-made knowledge management projects ranging from the Dublin Core metadata to Gene Ontology. The inter-operability of the tool is facilitated by making the knowledge model compatible with the Open Knowledge Base Connectivity (OKBC) protocol because of that the users can import and export Ontologies [12]. The knowledge model is frame-based and constitutes the building blocks of the ontology. It consists of classes that support multiple inheritance slots that attribute properties to classes, facets that describe slots, axioms that register constraints and instances of classes that specify values for slots. Protégé-2000 is shown in Figure 3.

6.2 OILEd

OILEd [13], is a Java based ontology editor developed by the University of Manchester, UK. The knowledge model of OILEd is an extension of the frame-based system and, unlike Protégé-2000, allows the use of highly expressive languages. However, it does not support the development of large scale Ontologies and the versioning, migration and integration of

Fig. 3. An example of Protégé-2000.

Ontologies. The tool contains concepts for the domain of discourse as classes, slots describing the properties of classes, the constraints as axioms and instances. A diagram representing the editor is illustrated in Figure 4. The languages that are supported by the editor besides DAML+OIL are RDF, RDFS and SHIQ [14] for classification purposes.

A major feature of OILEd is the FACT Reasoner [15], a CORBA interface written in LISP that checks the consistencies of Ontologies by translation from DAML+OIL to the SHIQ DL.

6.3 OntoEdit

OntoEdit [16], it is a graphical based Ontology Engineering tool that supports the development and maintenance of Ontologies and is available either in the stand-alone or client-server format. The internal ontology model on which it is based allows the domain to be developed through the use of classes, relations, axioms, facets and attributes. The interoperability of the tool is facilitated by the development of plugins, which permits it to interact with other tools like Ontobroker and Sesame. There are plugins that allow the user to import and export Ontologies, portal assembly kits, integrated inferencing and ontomapping. The tool permits the usage of DAML+OIL and RDFS. There are two versions available. *The Free version* is limited to deal with only 50 instances whereas the

Fig. 4. An example of OILEd 3.4.

Professional version, in addition to the facilities available for the Free version, has an extended functionality that allows for the checking of consistency of the ontology and an ontology server for the administration and storage of Ontologies.

6.4 Implementation Ontology Tools Summary

We show in Table 1 a summary about ontology building tools, some web-based tools for making Ontologies are mentioned too.

7 Parsers and Database Technology of the Semantic Web

Parsers and repositories provide support and additional functions or services like: accessing, queries, storage, etc. to the systems developed that work with knowledge resources.

7.1 RDF Parsers

RDF parsers provide the basic support in parsing different RDF serializations, accessing RDF triples via programming interfaces or queries, and providing basic operations with the triples. The SiRPAC1 RDF parser. SiRPAC is a set of Java classes that can parse RDF/XML documents

Product	URL	Company	Description
OilEd	http://oiled.man.ac.uk/	University of Manchester	OilEd is an ontology editor allowing the user to build Ontologies using DAML+OIL.
OntoEdit	http://www.ontoknowledge.org/tools/ontoedit.shtml	AIFB, University of Karlsruhe	OntoEdit is an Ontology Engineering Environment supporting the development and maintenance of Ontologies using graphical means. OntoEdit is built on top of a powerful internal ontology model. The internal ontology model can be serialized using XML, which supports the internal file handling.
Ontolingua	http://www-ksl-svc.stanford .edu	KSL of Stanford University	Web-based tool for making Ontologies.
OntoSaurus	http://www.isi.edu/ isd/ontosaurus.html	ISX	It is a web browser for Loom knowledge bases. Also, it is possible to browsing and editing Ontologies.
Protégé-2000	http://smi-web.stanford.edu/projects /prot-nt/	Stanford's Medical informatics Section	Tool for building Ontologies of domain models, and it has been developed to assist software developers in creating and maintaining explicit domain models, and in incorporating those models directly into program code.
WebODE	http://delicias.dia.fi.upm.es/ webODE/	Universidad Politécnica de Madrid	WebODE is a workbech for ontological engineering that not only allows the collaborative edition of Ontologies at the knowledge level, but also provides a scalable architecture for the development of other ontology-based applications.
WebOnto	http://webonto.open. ac.uk/	Knowledge Media institute of the Open University	It was designed to support the collaborative browsing, creation and editing of Ontologies.

Table 1. Summary of tools for building Ontologies

Fig. 5. OntoEdit main window

into the three-tuples of the corresponding RDF data model. The parser has evolved in several versions and has become a *de-facto* standard in Java-based RDF development.

7.2 RDF Repositories

RDF repositories are built on top of RDF parsers and provide additional functions concerning storage of large amount of triples in relational databases, reformulation of RDF queries to SQL database queries, providing efficient RDF parsing and interchange functions. The rdfDB project [17] is intended to be a simple, scalable, open-source database for RDF. The rdfDB repository, which is now under intensive development, provides the following basic RDF database operations: database creation, adding triples into a database, namespace manipulation, database querying. Query interface supports only one query command, which allows to retrieve a set of triples which satisfy a certain set of constraints. The support for RDF Schemas and inference are planned, but not yet designed or implemented.

7.3 Sesame

Sesame [18] is an RDF schema-based repository and querying service, being developed by Administrator in the frame of the On-To-Knowledge project. Sesame is a system that allows persistent storage of RDF data and schema information and subsequent online querying of that information. Sesame has been designed with scalability, portability and extensibility in mind. One of the most prominent modules of Sesame is its query engine. This query engine supports an OQL-style query language called RQL. RQL, supports querying of both RDF data and schema information. RQL also supports path-expressions through RDF graphs, and can combine data and schema information in one query. Sesame provides a substantial functionality in querying RDF instances and RDF Schemas, using an object oriented query language RQL. The Ontologies and their instances are accessible in a uniform way by means of RQL queries. The RQL language implemented in Sesame allows querying class definitions and class instances by their properties, supporting an extensive set of querying expressions.

8 Other Tools that Generate Semantic Content

Other Semantic Web tools and Services include:

CORPORUM OntoBuild [19], which extracts content representation models from natural language texts. CORPORUM consists of two parts: CORPORUM-OntoExtract and CORPORUM-OntoWrapper. Both are under development and are considered to be alpha stage software.

Fast Classification of Terminologies (FaCT) [15] description logic classifier.

OntoBroker [20], which includes a Frame-Logic-based inference engine, a crawler component, and several ontology-based semantic annotation tools.

WebScripter [21] is a tool for assembling reports by extracting and fusing information from multiple, heterogeneous DAML-based web sources.

OntoMat-Annotizer [22] is a Java-based annotation tool that lets users enrich web pages with DAML+OIL.

Semtalk [23] reads and writes a subset of RDFS storing RDFS models as hyperlinked XML flat files and uses Microsoft's Visio to model information graphically. Semtalk's integrated object engine also handles object representation and some reasoning.

9 Ontology Examples

These examples are categorized into two different groups: *domain specific Ontologies* (UMLS, GO, SWRC) and *domain general Ontologies* (WorNet). Each ontology example is described in the Table 2. Some examples of these Ontologies are presented in the Figures 6 and 7.

Fig. 6. An example of the Gene Ontology.

References

1. Javier Gramajo. State of the art: Ontologies. 2005.
2. T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
3. B. Park. Domain modeling for knowledge-based systems: An approach based on objects, events, and rules. Technical Report 1992, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1997.
4. D. Fensel. Ontologies: Silver bullet for knowledge management and electronic commerce. *Springer-Verlag, Berlin*, 2001.

Fig. 7. An example of SWRC Ontology.

Ontology	URL	Organization	Description
UMLS	http://umlsks.nlm.nih.gov	US National Library of Medicine	The Unified Medical Language System (MLS). It allows users to draw information from several medical databases like MEDLINE and Read Codes. It has 730,000 concepts, 130,000 entries and 134 semantic types.
GO	http://www.geneontology.org		The consortium of Gene Ontology [24] produces dynamically controlled vocabulary that can be applied to all organisms even as the knowledge of proteins and genes changes. The Ontologies are developed based on three principles: a) molecular function, b) biological processes, c) cellular component. It also maps other biological databases like SWISS-PROT, from Enzyme Commission to those of the GO terms. A search for the 1,4-alpha-glucan branching enzyme displayed in the Figure 6.
SWRC	http://ontobroker.semanticweb .org/ontos/swrc.html	University of Karlsruhe, Germany	The Semantic Web Research Community (SWRC) Ontology it models the research community of the Semantic Web and annotates the publications, the tools, researchers and the relations among them. A SWRC ontology is shown in the Figure 7.
WordNet	http://www.cogsci.princeton .edu/ wn/index.shtml	Cognitive Science Laboratory at Princeton University	Online lexical system that outlines the semantic relations between word meanings. It categorizes the English words according to nouns, verbs, adjectives and adverbs. These English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets.

Table 2. Ontology examples summary

5. T. Berners-Lee. Weaving the web: Ontology infrastructure for the semantic web. *Orion Business Books*, WonderWeb:1, 1999.
6. J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems Journal*, 16(2), March/April 2001.
7. *XML, The Extensible Markup Language*. http://www.w3.org/XML/.
8. A. Maedche. Development and applications of ontologies. *ECML/PKDD*, Tutorial, 2001.
9. *The WebsScripter*. www.isi.edu/licensed-sw/webscripter.
10. J. Heflin and J. Hendler. Dynamic ontologies on the web. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 443–449, Menlo Park, CA, July 30– 3 2000. AAAI Press.
11. *The Web Ontology Language (OWL): Overview*. http://www.w3.org/TR/owl-features/.
12. *The Protégé*. http://protégé.Stanford.edu.
13. C. Goble S. Bechhofer, I. Horrocks and R. Stevens. OilEd: A reason-able ontology editor for the semantic Web. *Lecture Notes in Computer Science*, 2174:396, 2001.
14. I. Horrocks and U. Sattler. Optimised reasoning for SHIQ. In *Proc. of the 15th European Conference on Artificial Intelligence*, 2002.
15. *The Fast Classification of Terminologies (FaCT)*. http://www.cs.man.ac.uk/ horrocks/FaCT/.
16. *The OntoEdit*. $http://www.ontoprise.de/download/ontoedit_data_sheet.pdf$.
17. *rdfDB Project*. http://www.guha.com/rdfdb/.
18. *The Sesame*. http://sesame.aidministrator.nl/.
19. *The CORPORUM OntoBuild*. http://ontoserver.cognit.no/index.html.
20. *The OnToBroker*. $http://ontobroker.aifb.uni-karlsruhe.de/index_ob.html$.
21. *The WebOnt*. www.w3.org/2001/sw/WebOnt/.
22. *The OntoMat-Annotizer*. http://km.aifb.uni-karlsruhe.de/annotation/
23. *The SemTalk(tm)*. http://www.semtalk.com/.
24. *The Gene Ontology (GO)*. http://www.geneontology.org.