Managing Ontologies: A Comparative Study of Ontology Servers

Mohammad Nazir Ahmad and Robert M. Colomb

School of Information Technology and Electrical Engineering
The University of Queensland
AUSTRALIA

{nazir,colomb}@itee.uq.edu.au

Abstract

An ontology is increasingly becoming an essential tool for solving problems in many research areas. The ontology is a complex information object. It can contain millions of concepts in complex relationships. When we want to manage complex information objects, we generally turn to information systems technology. An information system intended to manage ontology is called an ontology server. The ontology server technology is at the time of writing quite immature. Therefore, this paper reviews and compares the main ontology servers that have been reported in the literatures. As a result, we point out several research questions related to server technology.

Keywords: Ontology, Ontology Server, Ontology Tool.

1 Introduction

An ontology is an explicit specification of a conceptualization (Gruber, 1993). It is a designed artefact that formally represents agreed semantics of a domain interest in computer resources (Gruber, 1993; Guarino, 1998). This enables the sharing and reuse of information and allows for the interoperation of information systems (Pretorius, 2005). Although not a new field, ontology research has recently received renewed interest in many fields such as semantic web (e.g., Berners-Lee, 1999; Mika, 2004), databases (e.g., Guarino, 2002), electronic commerce (e.g., Barley et al., 1997; Corcho et al., 2001), knowledge management (e.g., Rothenburger et al., 2006; Chi et al., 2006), Khosla, 2003), information retrieval (e.g., Guarino et al., 1999; Abdelali et al., 2003), bioinformatics (e.g., Karp et al., 2002; Bada et al., 2004), software engineering (e.g., Welty et al., 1999; Derridder et al., 2000), intelligent systems (e.g., Akkermans et al., 2004) and so forth. Ontology applications have been classified in Jasper et al. (1999), Mizoguchi (2003) and most comprehensively survey proposed by Hart et al. (2004).

2 Managing Ontologies and Ontology Tools

A major problem with ontologies is how to manage them. Thus, there are related research problems in the field of ontologies which include: (a) creating large-scale ontologies (e.g., Lenat et al., 1995), (b) defining expressive languages for representing ontological knowledge (e.g., Lopez et al., 1999; Karp et al., 1999; Fensel et al., 2001; Bechhofer et al., 2004; Klyne et al., 2004; Brickley et al., 2004) and (c) implementing systems or tools that support ontology-based applications (e.g., Farquhar et al., 1997; Chung et al., 2000; Cui et al., 2000; Li et al., 2003; Dameron et al., 2004; Valo et al., 2005; Starlab, 2006).

When a new ontology is going to be developed, several basic questions arise related to the methodologies, tools and languages to be used in ontology development as reported in (Corcho et al., 2003). However, the management tasks in ontologies are not ontology development, but it should focus on other aspects such as how to use ontologies for example in supporting information systems interoperation. We would say these aspects are from two fundamental ontology engineering challenges: reusability and usability. The former aspect is main goal in building ontologies (Uschold et al., 1996; Motta et al., 1999; Gomez-Perez et al., 1999; Gruber, 1995). The latter aspect is how to make use of ontologies. This is an important goal in interoperating systems, where an exchange can involve thousands of players. We believe that most research focuses on the development of single or integrated tools for supporting ontology development. In other words, they aim to support ontology reusability. The main ontology tools in the mainstream literature for ontology development are well reviewed in (Corcho et al., 2003; Duineveld et al., 2000; Mizoguchi, 2003; Fensel, 2001).

We believe that the notion of ontology server has originally stemmed from the research of ontology development tools. Most works describe implicitly or explicitly the server as a kind of isolated or integrated tool for building ontology (e.g., Li et al., 2003; Farquhar et al., 1997; Eklund et al., 2002; Noy et al., 2002; Sure et al., 2002; Noy et al., 2000). The isolated tools are not fully integrated with other activities of the ontology lifecycle (Corcho et al., 2003). Such tools are Ontolingua Server (Farquhar et al., 1997), Ontosaurus (Swartout et al., 1997) and OntoRama (Eklund et al., 2002). The latter is much more ambitious, built as robust integrated environment or suite that provides technological support to most of the ontology lifecycles activities. They have extensible, component-based architectures, where modules can easily be added to provide more functionality to the environment. (Corcho et al., 2003). Among these environments, we can cite Protégé-2000 (Noy et al., 2000), WebODE (Arpirez et al., 2001) and OntoEdit (Sure et al., 2002).

However, the discussion of ontology server in those ontology development tools is quite confusing and still unclear. Some studies discuss the server as ontology repository. In this context, the server is mainly about database technology (e.g., Pan et al., 2003; Harrison et al., 2005). Some other studies discuss both ontology repository and server functionality. In this context, the server is described as an information system (e.g., Farquhar et al., 1997; Li et al., 2003; Starlab, 2006; Mauger, 2005; Dameron et al., 2004; Chung et al., 2000; Swartout et al., 1997; Arpirez et al., 2001; Sure et al., 2002). Therefore, we would like to highlight three important points about the notion of ontology server discussed in many studies as follows: (a) many studies discuss ontology tools mainly for building ontology. Some of these tools consist of many other modules including the ontology server. In this context, the notion of ontology server is mostly a database issue. For example, Corcho et al. (2003) describe Ontosaurus (Swartout et al., 1997) as consisting of two modules: ontology server and ontology browser. In other words, the notion of ontology server is only referred to ontology repository such as in (Pan et al., 2003; Harrison et al., 2005). (b) Some studies discuss clearly ontology servers and use the word "ontology server" in their work. These studies include both server ontology repository and functionalities (e.g., Li et al., 2003; Starlab, 2006; Mauger, 2005; Dameron et al., 2004; Oberle et al., 2002), (c) In some other studies are similar to (b), except the notion of ontology server is limited to ontology repository as mentioned in (a). An example is the Ontolingua server.

We draw three conclusions for research directions form understanding in three important things: First the points above, we argue with the notion of ontology server as mentioned in (a) and (c). The central issues in ontology server development should include ontology repository and server functionalities. Further, an ontology server is a kind of information system and we can discuss the server from information systems perspective. Second, most of the ontology servers in the literature are used for building ontologies. In other words, the current studies give emphasis more the ontology server supporting some activities in the ontology building lifecycle. We argue that, third, to have a clearer understanding in this area, we would need a so-called "framework" to give a brief idea of ontology server research.

To extend these positions, we point out several basic questions related to the functionalities, repository, and methodological support and so forth to be used in the ontology server development. These questions would include:

> *What are the main components of ontology server? What kind of framework can we use to understand the server? What methodologies can we use for building an ontology server? Does any methodology provide support for building the server? What general activities should we consider when building the server? Which is the life-cycle of an ontology that the server can support? How are the ontologies stored in the server? What are the main server functions? How can applications interoperate with ontology servers and/or use the ontologies? What sort of ontology languages is the server supposed to support? Is the language chosen appropriate for exchanging message between different applications? What is the main purpose of ontology server to be developed? What sort of the ontology server architecture can we propose? And so forth.*

In this paper, we will present the main characteristics of ontology servers, which can assist practitioners and researchers in this field to gain answers to the questions above. Since at the time of writing this field is fairly immature, we will provide guidelines that outline several opportunities related to ontology server research.

3 Ontologies and Semantic Web

Ontologies are an important part of the semantic web (Berners-Lee, 1999). The semantic web is an extension of the current web in which information is given well defined meaning, better enabling computers and people to work in cooperation (Berners-Lee, 1999). In the semantic web, ontologies can be used to encode meaning into a web page, so that intelligent agents (applications) can understand what the web page is about, and therefore provide humans with more useful cooperative services (Berners-Lee, 1999). Driven by application needs and the semantic web vision, the ontology server is believed to be a key component for supporting semantic web applications (Agrawal, 2002). Therefore, in this study, we will only focus on the ontology servers that support ontology-based applications in open environments such as the web and semantic web. Furthermore, we will look into several ontology servers: (a) are mostly cited in literatures, (b) they explicitly use word "ontology server" in their reports and address server development issues, (c) we assume that several of them are still progressing and (d) focusing on two main aspects of the ontology server: server functionalities and ontology repository.

4 An Ontology Server

Ahmad et al (2006) show that the ontology server is used at design, commit and runtime. At design time, the ontologist uses the server for developing ontologies. At commit-time, players (i.e. applications) will commit to a limited part of the ontology to enable the exchange of messages. At runtime, players exchange messages mediated by shared ontologies. Zachman (1987) has proposed an information system architecture framework, which has been well accepted (Evernden, 1996; Inmon et al., 1997; Ells, 1998; O'Rourke et al., 2003; Jones, 2005). There are three key concepts (knowledge, process and communication) and related enabling technologies described by Zachman (1985). Since the ontology server is a kind of information system, it can be described in terms of this framework (see details in (Ahmad, 2006)). In the context of server development, it is firstly important to concentrate on the process aspect (server functionality) and knowledge aspects (ontology repository) of ontology server design.

4.1 Ontology Server Functionality

There is a number of different ontology server implementation available. Their functionality focuses on editing, browsing and storing ontologies. In some cases the ontology server also provides an inference engine that allows statements about the relationships between entities in different ontologies to be tested or retrieved (Li et al., 2003). Pan et al (2003) state a number of facilities the server may provide such as support for creating and editing ontologies, support for publishing and retrieving ontologies (by humans via a graphical user interface and/or via a network protocol), support for recording metadata about ontologies and the relationships between them, support for interactively browsing the structure of ontologies and inference mechanisms to verify the consistency of ontologies.

We would say that there are two main approaches to implementing ontology server functionalities: (a) tool development and (b) Application Programming Interfaces (APIs). The former approach is about developing any kind of application on top of the ontology repository such as ontology browser, ontology editor, ontology translator and so on. In this approach, those tools typically perform a single aspect of ontology server functionality. Those tools can make use of an ontology stored in the ontology repository. For example, Starlab (2006) has developed an ontology browser, ontology manager and ontology modeller to make use ontologies stored in the server repository. The API approach is to write a program that provides some services to the ontology repository. Typically, any applications or tools can use those APIs to interface the ontology repository. For example, Starlab (2006) has provided some basic database APIs for accessing ontologies stored in the ontology repository. With respect to server functionality, to avoid confusion, we should be aware of many terms appearing in studies of server functions such as server functionality and functions (Starlab, 2006), ontology service (FIPA, 2001), server process (Chung et al., 2000), server services (Farquhar et al., 1997), server operation and ontology operation (Dameron et al., 2004), and server facilities (Pan et al., 2003).

4.2 Ontology Server Repository

In this paper, we use term “ontology repository” and “server repository” interchangeably. We would say that the discussion of ontology repository gives emphasis more to database issues such as storing and organizing ontologies in database. There are such studies discussing specifically ontology repository aspects such as Pan et al. (2003) and Harrison et al. (2005). For example, Harrison et al. (2005) have proposed a generic representation for ontology repository. Similar to Pan et al. (2003), they have introduced a so-called lightweight ontology repository for enabling shareable and maintainable ontologies. A lightweight ontology is referred to a kind of ontology that simple, generic form and does not include axiom that allow deductions to be made. Repository is a prototype implementation of a design to allow ontology designers and agents to use open web standards to publish and retrieve ontologies and metadata about them. Key features of the system include the use of the HTTP protocol following the REST (Representational State Transfer) architectural style, the representation of recorded information about ontologies and the repository information schema using RDF and its schema language RDFS, and the use of URNs to identify ontologies. The use of web standards for communication between agents and web-based resources such as ontology repositories enables a more lightweight and open architecture for agent interaction with these resources (Pan et al., 2003). Harrison et al. (2005) discuss two methods of storing the ontologies. The first method involves storing individual ontology in a separate flat file. The file provides a more straightforward view where its content can be inspected. However, the main problem with this method is that a search engine would need to be developed to search the contents of the ontology files (Harrison et al., 2005). The second method uses a database to store ontologies. Databases have indexing and other capabilities that enable faster searching. Therefore, to ensure scalability and maintainable large ontologies, the database method rather than a flat file we think would preferable for storing ontologies. However, how the ontologies are stored depends on how the ontology representation should appear in the server and what types of database are considered (i.e. relational, object-oriented or object-relational).

4.3 A Comparison of Ontology Servers

After an extensive search on the Internet and several journals and conferences, and with several focus as stated in section 3, we selected the following eight ontology servers: Ontolingua server (Farquhar et al., 1997), ACOS (Li et al., 2003), Starlab (2006), KAON (Oberle et al., 2002), OntoRama (Eklund et al., 2002), OWS (Dameron et al., 2004), FIPA server (2001), and Adapted Ontology Server (Chung et al., 2000).

The Ontolingua server has been running since 1995. It has been developed in the Knowledge Systems Laboratory (KSL) at Stanford University. It is a tool that supports distributed, collaborative editing, browsing and creation of Ontolingua ontologies with a form-based web application (Corcho et al., 2003). Remote editors can browse and edit ontologies, and remote or local applications can access any of the ontologies in the ontology repository with the OKBC protocol (Farquhar et al., 1997). The Ontolingua ontology uses the representation languages, Ontolingua Frame Ontology and KIF, which are wide spectrum language capable of representing fine features of concepts (i.e. are based on description logics). The Ontolingua server supports ontology inclusion and circular dependencies (Farquhar et al., 1997). Its consistency-check capability, however, is restricted to the functions similar to database schema checking (Li et al., 2003). For instance, “all slots, slot values, facets and facets values are checked to make sure that they conform to the constraints that they apply (i.e. domain, range, slot value type, and cardinality constraints)” (Li et al., 2003). This means that semantic consistency checking is done in domain experts’ heads. It is a large project focusing on ontology development (Cui et al., 2000). It has built sophisticated tools for developing and maintaining frame-based ontologies. It focuses on formal ontology specifications, and reuse and translation to different ontology implementation systems (Cui et al., 2000). However, it does not address problems related to legacy systems and tools to merge ontologies (Cui et al., 2000). The users interact with the single server through a web browser to create, edit and browse ontologies. Users have to tolerate the network delays and server response delays. The Ontolingua server uses Ontolingua-based repository for storing ontologies and developing a set of tools for demonstrating server functions.

The second server is ACOS. It has been developed in the Intelligent System Laboratory (ISLab) at British Telecom Research. Li et al. (2003) claim it to be a community-oriented ontology server. It provides the way for the so-called "community" to construct ontologies. The concept of "community" in ontology management enables everyone to have the opportunities of influencing ontology construction. This depends on the users' importance score, and such score is computed mechanically based on how active these users are in contributing to the knowledge base. They came up this kind of vision because in an open environment, ontology is the asset of all participants; every user can join, contribute and leave such community. The opposite of this approach is called "central". The central-controlled mechanism would not appear to be appropriate for this situation (Li et al., 2003). However, most of the ontology server implementations in many studies apply the "central" approach to ontology development (e.g., Farquhar et al., 1997; Chung et al., 2000; Frehiwot et al., 2001; Eklund et al., 2002; Starlab, 2006). The "central" means a centralized ontology server, only a fixed group of users have the rights to modify ontologies, which is similar to the situation in database management (Li et al., 2003). Similar to Ontolingua server, the ACOS has also provided consistency checking but not limited to humans. A wide range of software agent communities also can share it and the server facilitates on-line ontology construction, consistency-check and use (Li et al., 2003). Li et al. (2003) claim that ACOS is designed to be an online community in which a diverse group of software agents can contribute and use ontologies at runtime. A key enabler in this scenario is a high degree of "shareness" of the ontologies maintained by the server. This relates to designing an appropriate knowledge representation, which is the first step towards building an ontology server. In this context, Li et al. (2003) believe that to achieve "shareness", it is required that local features of ontology are removed. Examples of constructors that can bring in local features include *property* in DAML-OIL or *slot* in Ontolingua (Farquhar et al., 1997). Other examples of constructors that bring in local effect are *part-of* in Framework (Farquhar et al., 1997) and *disjoint-with* in Descriptions Logic (Borgida et al., 1989). In other words, Li et al. (2003) accept the shared ontologies should have minimal expressiveness, which consist of a minimal set of axioms written in a language of minimal expressivities. Thus, to achieve a high degree of shareness, Li et al. (2003) believe that constructors *Class*, *subClassOf*, *SameClassAs*, *SuperClassOf* and *InstanceOf* offer more stability in the process evolution and minimal expressivities. Compared to Ontolingua server, the ACOS uses DAML-OIL as its ontology representation language. The ACOS server uses file-based repository for storing DAML-OIL ontologies and developing a set of APIs for implementing server functions. To facilitate collaborative ontology development, this server implemented an import mechanism that is similar to the *inclusion model* in Ontolingua server and *extension relationship* in FIPA ontology server (FIPA, 2001).

Third is the Starlab. This server is under development and still progressing several deliverables. This research initially falls within the DOGMA research framework (Jarrar et al., 2002). At the time of writing, we would say that this is a new ontology server research project, is funded for 5 years by the Vrije Universitet Brussel. The mission of this server is to assist the gathering and incremental growth of ontologies (Starlab, 2006). In terms of ontology representation, this server is in line with the rationale of ACOS server previously discussed. The proposed ontology model consists of five basic elements: context, terms, concepts, roles and lexicon (Starlab, 2006). Starlab (2006) claims that constraints and derivation rules are intentionally left outside the ontology. At this moment, they are still experimenting with the ontology model and an early version of ontology server has been implemented. In the first prototype, consistency-check and user control are not included (Starlab, 2006). This server uses MSQL server to store the ontology and ontology objects are expressed in XML. Its server functions are implemented based on developed tools and basic database APIs (Starlab, 2006). In addition, the server's ontology representation also benefits from graphical notation based on Object Role Modeling (ORM) languages.

Fourth is the KAON server developed by AIFB in Karlsruhe University. Similarly to Ontolingua server and Starlab, the KAON server is a result of sophisticated tools delivered by a large research project conducted at the University of Karlsruhe. This server provides its functions through a set of APIs and store RDF-based ontologies using a relational database (Oberle et al., 2002; Maedche et al., 2003). It is an early prototype of an ontology-based application server. In essence, it uses JBoss and extends by the tools of the KAON tool suite for reasoning with software components (Oberle et al., 2002). The goal is to support the developer in his daily tasks with reasoning. In terms of ontology representation, it is similar to Ontolingua server: the complexity of ontology is not left outside the ontology, which contrasts with Starlab and ACOS.

Fifth is the OntoRama server was developed by Eklund et al. (2002). The main functions of OntoRama include search, compare and modify WebKB ontologies. It does not support consistency-check or cross-ontology queries; as a result, its capability of supporting online collaborative ontology construction is restricted (Li et al., 2003). In our opinion, this server demonstrates functions of storing and browsing RDF-based ontologies, while supporting ontology development is not its main purpose.

Sixth, the OWS server proposed by Dameron et al. (2004) at the Stanford Medical Informatics (SMI), Stanford University. This server does not generally deal with ontology development. It is used for all other tasks, assuming ontologies already exist. The implementation of this server is bit tricky. In fact, the server described by Dameron et al. (2004) does not itself serve as ontology repository (i.e. the server itself does not store the ontologies). The server provides services, taking ontologies as inputs. Hence, they claim that ontology evolution is not a problem here, because a change simply gives the server a new ontology as inputs. In our opinion, we would say that this server supports runtime interoperation. Server functions are implemented using web services assuming that ontologies naturally reside in any player's sites. In other words, the ontology storage is file-based, which are held on different players' sites.

Seventh is the FIPA ontology server. The FIPA (Foundation for Physical Agents) is a standards body, which has developed interaction standards for agents in open environment. In the FIPA's ontology server specification, its server's ontological representation is divided into a *fine-grained ontology,* called heavyweight ontology in (Pretorius, 2005) and *a coarse ontology* that consists of a minimal set of axioms written in a language of minimal expressivity, called lightweight ontology in (Pretorius, 2005). However, the FIPA's server supports these two kinds of ontology representation but with different scope of use and level of detail (FIPA, 2001).

Eighth is the server proposed by Chung et al. (2000) at SungKyunKwan University, Korea. This server is intended specifically for ontology developed for electronic commerce applications. In terms of ontology representation, Chung et al. (2000) define two main criteria for ontology: (a) ontology can be translated and (b) ontology should be practical. Its server functions include gathering information from the web, creating a relation, modifying and rebuilding the standard ontology and servicing the standard ontology. This server uses a MySQL database to store ontologies and Java APIs for implementing its server functions (Chung et al., 2000). We assume that this server allows medium ontological representation. This is because the standard ontology is built from the local terms used in sites, and then the server provides an editor tool for making relation between ontologies (Chung et al., 2000). In addition, they claim that a standard ontology necessarily has the objective and concrete property (Chung et al., 2000). In general, this server supports ontology development but is limited to electronic commerce and deals with the current web.

4.4 Comparison Framework

A survey of ontology servers was firstly done in (Li et al., 2003). However, Li et al (2003) did not include several key ontology servers such as KAON and OWS in their survey. In addition, the proposed comparison is too general and does not highlight other important aspects of development issues such as a standard development approach, methodological support and the use of graphical modeling language. In this study, we have compared ontology servers with respect to the similar criteria to (Li et al, 2003), extending some of them, also adopting several dimensions used in (Duineveld et al., 2000; Corcho et al., 2003). Consequently, we incorporate all these criteria into five dimensions as follows:

General: It refers to the generic aspects of the server development such as: *What is the main goal of developed ontology server? Who are those developers? Can we access to those servers? What is the main phase that server can support? What are the main deliverables of the server development? Which kinds of management provision does the server have? Does the server support collaborative environment? What is the status of the server?* **Ontology:** This dimension refers to the general questions about ontology representation that the server supports, such as: *What the main ontology engineering challenges that the server mostly focuses on? What types of ontology can the server support? What does knowledge representation in server repository look like? What ontology language is the server based on? How expressive is language? Does the server use all the language features (i.e. ontological constructors) to represent shared ontologies?* **Features:** It represents general features of ontology server in terms of: *What are the main server functions have been developed? What types of repository platform the server use to store ontologies? What is the architecture of the server? Does the server provide extensibility?* **Implementation:** This dimension addresses some aspects of the server technology including: *How is the server functionality implemented? What type of technology platform does the server use and support? Does the server use a standard technology? What type of technology is used to access to the ontology repository?* **Methodology:** This dimension concerning the methodology that ontology server gives support to, and also the methodology used to develop a server such as: *Is there any methodology support for building the server? Is there any methodology support for the ontology?* To have a clearer understanding of this study, as a result, we summarize this comparative study in Table 1 (see appendix).

5 Conclusions and Future Works

We point out some important points based on a Table 1. First, we see that most ontology servers available mostly focus on design time (see *main phase of ontology life cycl*e). It gives supports more to ontology development and tends assume its users to be ontologists. Its main goal is reusability (see *main ontology engineering challenges*) since it saves time and cost. From this, we argue that the server development should be determined by what ontology is made for. For example, the ontologies supporting runtime interoperation, which used in business application, are slightly different from ontologies for supporting engineering applications (Colomb et al., 2006; Hart et al., 2004). Therefore, we should firstly understand the intended use of the ontology which will then lead to how it is engineered and then how the supporting server is going to be developed. Second, surprisingly, none of the existing ontology servers are discussed in the context of commit-time issue (see *main phase of ontology life cycl*e). Although in general we have a design and runtime, but the issue how players can be helped to see the parts of ontology before joining the exchange should be firstly addressed. Third, a standard methodological approach for designing ontology and ontology servers is still missing (see *a methodology dimension*). Therefore, there is no server modelling profiles that can be referenced as useful example to guide a developer on how to develop ontology server. However, recently, there have been initiatives to bridge MDA (Model Driven Architecture) and ontologies. Many researchers suggest the use of Unified Modeling Languages (UML) in ontology development (e.g., Cranfield et al., 1999; Backlawski et al., 2002; Kogut et al., 2002; Colomb et al., 2006; Gasevic et al., 2005; Djuric et al., 2005). Thus, it would be advantageous to consider these works in the context of ontology and ontology server development. For example, it would be useful to benefit from the standard graphical ontological representations like UML. Unfortunately, most of the ontology and server developed do not consider the use of a standard graphical language.

Fourth, there is almost no explicit discussion of servers for supporting information systems interoperation in the mainstream literatures. So, the use of server as a runtime tool is not well developed. Thus, we would say, the server functions to support many aspects of the semantic web are still missing. Fifth, there are issues in ontological representation that relate to ontological repository. Most of the servers do not use database to store ontologies. However, we believe that most of the servers are moving towards using database and Java platforms (see *implementation*). Sixth, generally, there are two types of ontologies; lightweight and heavyweight (see *types of ontologies*). The former simply sees ontology as a description with the aim of organizing concepts. The latter defines ontology is a complete theory consisting of both a formal vocabulary and defined axioms that allow further deductions or inferences to be made. Most of the servers are influenced by lightweight ontologies. These servers only support using minimal language constructors, which have very minimal expressivity of ontology precision (i.e. *class, subClassof, IsA*). The resultant ontology from a knowledge representation point of view looks simple compare to heavyweight ontology. However, we argue that this situation is also determined by what the ontology is made for. For example, ontology supporting information systems interoperation needs heavyweight ontology that can be used by software agents.

From the perspective of a comparative study, we report the current technologies of ontology and the semantic web, particularly on ontology server development. Although we do not provide complete technical details, it is good enough to illustrate opportunities to enhance ontology server development. We know that ontology server is built around database issues but considering both server functionalities and server repository lead to treat the server is a kind of information systems that can benefit from other field like software engineering. Our future work is to investigate a useful methodological approach for designing ontology and ontology server in the context of commit-time issues. We argue that this should importantly be addressed since in the complex and large-scale ontologies, most of the players (i.e. software agents) are directly interested in the portion of the ontologies before joining some exchanges.

6 References

Ahmad, M.N., Colomb, R.M. and Cole, J. (2006): An Ontology Server: A Knowledge Tool For Systems Interoperability in Semantic Web Context. *Malaysian Journal of Information Technology,* 18(1): 1-23, June 2006. ISSN-0128-3790

Arpirez, J.C., Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A. (2001): WebODE: a scalable ontological engineering workbench. *Proc. First International Conference on Knowledge Capture (KCAP_01),* Victoria, pp. 6–13. ACM Press.

Agrawal, R. Making Semantic Web Real: Some Building Blocks. (2002): *Proc. International Workshop on Semantic Web,* Hawaii, USA.

Abdelali, A., Cowie, J., Farwell, D., Ogden, B., Helmreich, S. (2003): Cross-Language Information Retrieval Using Ontology. *Proc. TALN Batz-Sur-Mer,* France.

Akkermans, J., Baida, Z., Gordijn, J., Pe˜na, N., Altuna, A., Laresgoiti, I. (2004): Value Webs: Using ontologies to Bundle Real-World Services. *IEEE Intelligent Systems,* 19(4): 57–66.

Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Holmes, W., Letkowski, J., Aronson, M., Emery, P. (2002): Extending the UML for Ontology Development. *International Journal of Software and Systems Modeling,* 1(2): 142-156.

Barley, M., Clark, P., Williamson, K., Woods, S. The Neutral Representation Project. (1997): *Proc. AAAI'97 Spring Symposium on Ontological Engineering,* AAAI Press.

Borgida, A., Brachman, R.J., McGuiness, D.L. and Resnick, L.A. CLASSIC: A Structural Data Model for Objects. (1989): *Proc. ACM SIGMOD International Conference on Management of Data,* Portland, 58-67.

Bada, M., Stevens, R., Goble, C., Gil, Y., Ashburner, M., Blake, J. A., Cherry, M., Harris, M., Lewis, S. (2004): A Short Study on the Success of Gene Ontology. *Web Semantics: Science, Services and Agents on the World Wide Web,* 1: 235–240.

Berners-Lee, T. *Weaving the Web.* (1999): London, Orion Business Books, 1999.

Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuiness, D.L. Patel-Schneider, P.F. and Stein, L.A. OWL Web Ontology Language Reference. W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/. Accessed 23 July 2006

Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Holmes, W., Letkowski, J., Aronson, M. (2001): Extending UML to Support Ontology Engineering for the Semantic Web. *Proc. UML 2001,* Toronto, CA.

Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Holmes, W., Letkowski, J., Aronson, M., Emery, P.

(2002): Extending the UML for Ontology Development. *International Journal of Sofware and Systems Modeling* ,1(2):142-156.

Brickley, D., Guha, R.V. and McBride, B. RDFS Resource Description Framework Schema. W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/. Accessed 23 July 2006

Cranfield, S. and Purvis, M. UML as an Ontology Modeling Language. (1999): *Proc. Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence, IJCAI-99.*

Chung, H., Choi, J., Yi, J., Han, J., Lee, E. (2000): A Construction of the Adapted Ontology Server in EC, *Lecture Notes in Computer Science*, 1983:355-360.

Colomb, R.M., Raymond, K., Hart, L., Emery, P., Welty, C., Xie, G.T. and Kendall, E. (2006): Version 3.3: The Object Management Group Ontology Definition Metamodel. In *Ontologies for Software Engineering and Software Technology,* 1-25. Calero, Coral, Francisco and Mario, P. (eds). Springer.

Corcho, O., Gemez-Perez, A. (2001): Solving Integration Problems of e-Commerce Standards and Initiatives through Ontological Mappings. *Proc. IJCAI'01*.

Corcho, O., Fernandez-Lopez, M. and Gomez-Perez, A. (2003): Methodologies, Tools and Languages for Building Ontologies: Where is their meeting point? *Data & Knowledge Engineering,* 46: 41–64.

Chi, Y.L., Hsu, T.Y., Yang, W.P. (2006): Ontological Techniques for Reuse and Sharing Knowledge in Digital Museums. *Journal of Electronic Library,* 24(20): 147-159.

Cui, Z. and O'Brien, P. Domain Ontology Management Environment, *In the Proceedings of the 33rd Hawaii International Conference on System Sciences*, IEEE, 2000

Derridder, D., Wouters, B. The use of an Ontology to Support a Coupling between Software Models and Implementation. (2000): *Proc. European Conference on Object-Oriented Programming (ECOOP'00), International Workshop on Model Engineering.*

Dameron, O., Noy, N.F., Knublauch, H. and Musen, M.A. Accessing and Manipulating Ontologies Using Web Services. (2004): *Proc. Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications at the Third International Conference on the Semantic Web (ISWC-2004),* Hiroshima, Japan.

Djuric, D., Gasevic, D. and Devedzic, V. Ontology Modeling and MDA. (2005): *Journal of Object Technology*, 4(1):109-128.

Devedzic, V. Understanding Ontological Engineering. (2002): *Communications of the ACM,* 45(4): 136-144.

Duineveld, A.J., Stoter, R., Weiden, M.R., Kenepa, B. and Benjamins, V.R. (2000): WonderTools? A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies,* 52:1111-1133.

Eklund, P., Roberts, N., Gree, S. OntoRama: Browsing RDF Ontologies using a Hyperbolic-style Browser. (2002): *Proc. First International Symposium on Cyber Worlds (CW.02),* IEEE Computer Society.

Ells, R. Hypermedia, Help and How-To. (1998): *Proc. 16th Annual ACM SIGUCCS Conference on User Services*, California, United States, 363-368.

Evernden, R. The Information Framework. (1996): *IBM Systems Journal*, 35(1): 37-68.

Fensel, D. (2001): *Ontologies: A Silver Bullet for Knowledge Management and Electronic Management.* Berlin, Springer.

Farquhar, A., Fikes, R. and Rice, J. (1997): The Ontolingua Server: A Tool for Collaborative Ontology Construction. *International Journal of Human-Computer Studies,* 46:707-728.

Foundation for Intelligent Physical Agents (FIPA), FIPA Ontology Service Specification 2001, http://www.fipa.org/specs/fipa00086/index.html. Accessed 23 July 2006.

Gruber T.R. (1993): A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition,* 5(2): 199-220.

Gasevic, D.V., Djuric, D.O. and Devedzic, V.B. (2005): Bridging MDA and OWL Ontologies. *Journal of Web Engineering,* 4(2): 18-143.

Gruber, T.R. (1995): Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies,* 43(5/6).

Guarino, N. (1998): Formal Ontology in Informations Systems. *Proc. 1st International Conference on Formal Ontology in Information Systems (FOIS'98),* Trento, Italy.

Guarino, N., Masolo, C., Vetere, G. (1999): OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems,* 70-80.

Guarino, N. (2002): Ontology-Driven Conceptual Modelling. *Tutorial at 21st International Conference on Conceptual Modeling (ER'02),* Tempere, Finland.

Gomez-Perez, A., Benjamin, R. (1999): Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. *Proc. IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5),* Morgan-Kaufmann.

Hart, L., Emery, P., Colomb, R. M., Raymond, K., Chang, D., Ye, Y., Kendall, E. and Dutra, M. (2004): Usage Scenarios and Goals for Ontology Definition Metamodel. *Lecture Notes in Computer Science,* 3306:596. Springer.

Harrison, R and Chan, C.W. (2005). Distributed Ontology Management System. *Proc. 18th Annual Canadian Conference on Electrical and Computer Engineering, (CCECE'05),* Saskatoon, IEEE.

Inmon, W.H., Zachman, J.A., Geiger, J.G. (1997): *Data Stores, Data Warehousing, and the Zachman*

Framework: Managing Enterprise Knowledge. USA, McGraw-Hill.

Jasper, R. and Uschold, M. (1999): A Framework For Understanding and Classifying Ontology Applications. *Proc. IJCAI-99 Workshop on Ontologies and Problem-Solving Method (KRR5)*, Stockholm, Sweeden.

Jarrar, M and Meersman, R. (2002): Formal Ontology Engineering in the DOGMA Approach. *Proc. ODBASE'02,* 2519:1238-1254, LNCS.

Jones, S. (2005): Toward an Acceptable Definition of Service, *IEEE Software*,22(3): 87-93.

Karp, P.D., Chaundhri, V.K and Thomere, J. (1999): XOL: An XML-based Ontology Exchange Language. Version 0.3.

Karp, P.D., Riley, M., Saier, M., Paulsen, I.T., Paley, S., Pellegrini-Toole, A. (2002): The Ecocyc Database. *Nucleic Acids Research*, 30(1):56.

Klyne, G., Carroll, J.J and McBride, B. RDF Resource Description Framework Reference. W3C Recommendation, http://www.w3.org/RDF/. Accessed 23 July 2006

Kogut, P., Cranfield, S., Hart, L., Dutra, M., Baclawski, K., Kokar, M. and Smith, J. (2002): UML for Ontology Development. *The Knowledge Engineering Review*, 17(1): 61-64

Lenat, D.B. (1995): CYC: A Large-scale investment in knowledge infrastructure. *Communication of ACM*, 38(11):33-38.

Li, Y., Thompson, S., Tan, Z., Giles, N. and Gharib, H. (2003): Beyond Ontology Construction: Ontology Services as Online Knowledge Sharing Communities. *Lecture Notes in Computer Science*, 2870: 469 – 483, Springer-Verlag.

Lopez, M.F., Gomez-Perez, A., Sierra, J.P. and Sierra, A.P. (1999): Building a Chemical Ontology using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(1): 37-46.

Mika, P. Social Networks and the Semantic Web. (2004). *Proc. International Conference on Web Intelligence (WI'04),* Beijing, China, 285-291, IEEE Computer Society.

Mizoguchi, R. (2003): Tutorial on Ontological Engineering: Part 2: Ontology Development, Tools and Languages. *New Generation Comput.* 22(1).

Mauger, B. (2005): Tools to Help an Agent Commit to an Ontology. Undergraduate Thesis. Computer Science Department, University of Queensland.

Motta, E., Fensel, D., Gaspari, M. and Benjamins, R. (1996): Specifications of Knowledge Components for Reuse. *Proc. 11th International Conference on Software Engineering and Knowledge Engineering,* Kaiserslautern, Germany, 36-43, KSI Press.

Maedche, A., Motik, B. and Stojanovic, L. (2003): Managing Multiple and Distributed Ontologies on the Semantic Web. *The VLDB Journal*, 12(4):286-302.

Noy, N.F. and Fergerson, R.W. and Musen, M.A. (2000): The knowledge model of protege-2000: combining interoperability and flexibility. *Proc. 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW_00),* Lecture Notes in Artificial Intelligence, 1937:17–32. Springer.

O'Rourke, C., Fishman, N., Selkow, W. (2003): *Enterprise Architecture Using the Zachman Framework.* USA, Course Technology.

Oberle, D., Volz, R., Motik, B. and Staab, S. (2002): *KAON Server Prototype.* Technical Report, IST Project 2001-33052 WonderWeb, Department of Computer Science, The Victoria University of Manchester.

Pan, J., Cranefield, S. and Carter, D. (2003): A Lightweight Ontology Repository, *Proc. AAMAS'03*, Melbourne, Australia. ACM Press.

Pretorius, A.J. (2005): Visual Analysis for Ontology Engineering, *Journal of Visual Languages and Computing*, 16:359-381.

Rothenburger, B. and Galarreta, D. (2006): Facing Knowledge Evolution in Space Project: a Multi-Viewpoint Approach. *Journal of Knowledge Management*, 10(2): 52-65

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R. and Wenke, D. (2002): OntoEdit: Collaborative Ontology Engineering for the Semantic Web. *Proc. 1st International Semantic Web Conference*, Sardinia, Italy, Lecture Notes in Computer Science (LNCS 2342), Springer.

Swartout, B., Patil, R., Knight, K. and Russ, T. (1996). Towards Distributed use of large-scale ontologies. *Proc. 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

Star Lab. V.U.B. Ontology Server Research. http://www.starlab.vub.ac.be/research/dogma/OntologyServer.htm . Accessed 23 July 2006.

Uschold, M. and Gruninger, M. (1996): Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(2).

Valo, A. and Hyvönen, A and Komulainen, V. (2005): Tool for Collaborative Ontology Development for the Semantic Web. *Proc. International Conference on Dublin Core and Metadata Applications (DC 2005)..*

Welty, C. and Ferrucci, D. (1999): A Formal Ontology for Reuse of Software Architecture Documents. *Proc. International Conference on Automated Software Engineering,* 259-262, IEEE Computer Society Press.

Zachman, J.A. (1987). A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3):276-292.

(Appendix) Table 1. A Comparison of Ontology Servers

	Ontolingua	ACOS	Starlab	KAON	OntoRama	OWS	FIPA Ontology Service Specification	Adapted Ontology Server
General								
Main goal	Collaborative ontology development	Collaborative ontology development	Ontology development (i.e. gathering and incremental growth)	Ontology development	Storing, browsing ontology. Limited support to ontology development	Accessing and manipulating ontologies	Ontology development and ontology-based agent Communication	Ontology development specifically in electronic commerce
Developers	KSL	ISLab	STARLab	AIFB	ITEE and DSTC	SMI	FIPA	SECE
Release	Free web access	Not mentioned	Free web access	Open source	Free web access	No	Updated in 2001	No
Main phase of ontology life cycle	Design time	Design time	Design time	Design time	Design time	Runtime	Design time, Runtime	Design time
Main deliverables	Server functions mainly for supporting ontology development	Server functions mainly for supporting ontology development	Server functions mainly for supporting ontology development	Server functions mainly for supporting ontology development	Server functions mainly for browsing ontology	Server functions mainly for accessing and manipulating ontologies	Server functions mainly for ontology development and agent communication	Server functions mainly for supporting ontology development
Management provision	Central	Community	Central	Central	Central	Central – (Tasks)	N/A	Central
Collaborative (i.e. building, access and manipulating, use)	Yes	Yes	Not Mentioned	Yes	No	Yes	Not mentioned	Not mentioned
Overall remarks	Lack of extensibility, strictly oriented to research activities.	Lack of extensibility, strictly oriented to research activities.	Lack of extensibility, strictly oriented to research activities.	To support integrated environment and much more ambitious.	Lack of extensibility, strictly oriented to research activities.	Lack of extensibility, strictly oriented to research activities.	Provide standards ontology service that used as guidance in specifying ontology server functions.	Strictly oriented to research activities and do not support toward semantic web vision.
Status	Rolled out in 1995	Rolled out in 2002	Being Developed	Rolled out in 2001	Rolled out in 2002	Being Developed	Specification 2001	Rolled out in 2000
Ontology								
Main ontology engineering challenges	Reusability	Reusability	Reusability	Reusability	Viewing	Usability	Reusability	Extracting, Reusability
Type of ontologies	Heavyweight	Lightweight	Lightweight	Heavyweight	Lightweight	Heavyweight	Heavyweight, Lightweight	Lightweight
Knowledge representation	Complex	Simple	Medium	Complex	Medium	Open (unspecified)	Complex	Medium
Ontology model (i.e. expressiveness	Allow maximal expressiveness	Allow minimal expressiveness	Allow minimal expressiveness	Allow maximal expressiveness	Not mentioned	Not mentioned	Open (unspecified)	Not mentioned
Ontology languages	Ontolingua and KIF	DAML+OIL	XML	RDF	RDF	Open (unspecified)	Open (unspecified)	Not mentioned

Features								
Import mechanism	Yes	Yes	No	Yes	No	No	Yes	No
Manner of consistency checking	Human Approval, Simple Data Type Checking	Auto	Not mentioned	Yes	Not mentioned	Semi-auto	Open (unspecified)	Human
User access control	Yes	Yes	No	Yes	No	Yes	N/A	Not mentioned
Ontology repository platform	File-Based	File-based	DBMS	File-based + DBMS	File-Based	Open (unspecified)	Open (unspecified)	DBMS
Software architecture (i.e. client-server, n-tier application, etc)	Client-server	Client-server	Client-server	Client-server	Client-server	Client-server	Client-server	Client-server
Extensibility (i.e. plug-in features, etc)	No	No	No	Yes	No	No	Not Mentioned	No
Implementation								
Server functionality implementation	Tool development	APIs	Database APIs + Tool development	APIs + Tool development	Tool development	Semantic web services	Tool development	APIs
Server functionality platform	Not Mentioned	Java	Java	Java	Java	Java	Open (unspecified)	Java
User / Application remote support to ontologies (i.e. protocol used)	HTTP /OKBC	HTTP/ JDBC-ODBC	JDBC-ODBC	HTTP/JDBC-ODBC	HTTP/ JDBC-ODBC	HTTP/ JDBC-ODBC	HTTP/ OKBC	HTTP/JDBC-ODBC
Ontology repository technology	Ontolingua-based repository	Jena + DAML + OIL	MSQL Server	RDF + RDBMS	Jena + RDF	Open (unspecified)	Open (unspecified)	MySQL
Methodology								
Ontology-related methodology	No	No	Ontology double articulation and ontology modularization	No	No	No	METHONTOLOGY	No
Server development methodology	No	No	No	No	No	No	No	No
Benefit from graphical language	No	No	Yes – ORM for ontology modeling	No	Yes – Hyperbolic style for browsing ontologies	No	No	No
Server design profiles	No	No	No	No	No	No	No	No