Intelligence, Agents and Multimedia Group
School of Electronics and Computer Science
Faculty of Engineering and Applied Science
University of Southampton

Ontology Versioning and Evolution for Semantic Web-Based Applications

Yaozhong LIANG

yl504r@ecs.soton.ac.uk

A nine-month progress report submitted as per the requirements for
the continuation of study for the degree of MPhil/PhD

Supervised by Prof. Nigel SHADBOLT and Dr. Harith ALANI
{nrs, ha}@ecs.soton.ac.uk

July 2005

Abstract

Successful Semantic Web-based applications not only need large amounts of underlying well-organised and well-interrelated ontologies to support their infrastructures, but also need to provide the end-users with the consistent and continuous services as usual while the underlying ontologies changing. Research on ontology versioning and evolution addresses the issues of how ontologies cope with the internal and external changing environment so as to keep ontologies consistent. The existing works on ontology versioning and evolution focus on the changes inside ontology itself, i.e., the relationships and interoperability of the various versions, and largely neglected the issue of how changes during ontology versioning and evolution could affect the existing applications/services. The potential research work presented in this report aims at addressing how we could apply ontology versioning and evolution technologies in the existing applications so as to enable applications to provide consistent and continuous services as usual by adapting to the newly updating underlying ontologies. In this report, a middle layer between the underlying ontologies and dependent applications is proposed to build, which is used to monitor and detect any changes performed on the important parts of an ontology specific for the applications, and divert queries accordingly. The framework and requirements of the middle layer are discussed, in addition to a review of progress to date and anticipated future work.

Table of Contents

1 Introduction

1.1 Ontologies and the Semantic Web

1.1.1 Ontologies

Ontologies appear most effective when the semantic distinctions are crucial to the application's purpose. These applications span several areas including: Semantic Web research[1]; the creation of medical guidelines for managing patient health[2]; collaborative engineering design[3]; navigating, exploring and sharing works of arts through multimedia museum collections over the Web[4]; the automated exchange of electronic information among commercial trading partners, anti-terrorism intelligence analysis, command-level decision support in the battlefield, real-time military intelligence analysis, and so on.

As the hype of past decades fades, the current heir to the artificial intelligence legacy may well be ontologies [1] [3]. According to the report of Gartner[5], the market research firm, it identified that taxonomies/ontologies is one of the leading IT technologies, ranking it third in its list of the top 10 technologies [2]. In enterprises, Google and Yahoo!, the major web search services, are using ontology-based approaches to find and organize the contents on the Web. The join of Applied Semantics Inc., one of the leading vendors of semantic extraction tools, depicts a blueprint of the active role of Google in ontologies for their technologies.

1.1.2 The Semantic Web

Tim Burners-Lee first envisioned a Semantic Web which is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [7]. The Semantic Web is intended to provide machines with much better (automated) information access based on machine-processable semantics of data and heuristics that use these metadata so that they could be information intermediaries in support of humans [5][8]. It will combine together an incredibly large network of human knowledge which is organized by the ontologies, and will complement it with the power of machine processability.

In spite of the widespread use of the term "Semantic Web", it does not yet exist except in isolated environments, mainly in research labs. There are no exact agreements on what the Semantic Web is, nor what will it bring us. But from existing research works and investigations, representing human knowledge in ontologies to make it machine processable is a clear emphasis within the Semantic Web community.

---

[1] AKT IRC: http://www.aktors.org
[2] OpenGALEN: http://www.opengalen.org
[3] GEODISE: http://www.geodise.org
[4] The SCULPTEUR R&D Project: http://www.sculpteurweb.org
[5] Gartner, Inc. is the leading provider of research and analysis on the global IT industry. [http://www.gartner.com]

"The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purpose, but for automation, integration and reuse of data across various applications" [6]

1.2 Project Goals

Ontologies are regarded as the basic building units and integral parts for the Semantic Web, as they provide a reusable piece of knowledge about a specific domain. As the ontology development process becomes more ubiquitous and collaborative, the difficulties in maintaining different versions of ontologies become a serious problem. Failure to effectively maintain the versioning process of an ontology will hamper its reuse and deployment in external applications. Therefore, ontology versioning and evolution become essential and necessary task in ontology management. This is akin to the situations in software engineering where failing to maintain software codes and documentations may block the communications between different modules within a large project so as to hold down the performance of the application.

The research work presented in this report mainly brings up two questions for ontology versioning and evolution. The first question is how to keep record of changes performed between the various versions of an ontology so as to utilise the semantics bearing on the changes to make them traceable by any potential applications. The second one is how to manage the existing applications/services to provide continuous and updated services to the users when the underlying dependent ontologies have changed. This could be illustrated by a diverting system which aims to automatically divert any incoming queries from the applications/services adapt automatically to the recently updated underlying ontologies so as to deliver unchanged services to the users.

The ideas within this design are based on our investigation that most of the existing research activities are focusing on the relationships and interoperability between the various ontology versions. Research on updated ontologies within the application spectrum was omitted.

1.3 Outline of the Report

The rest of this report is organized as follows. In the next section, the motivations of to address the problems of ontology versioning and evolution will be stated. Section 3 introduces the necessary technical background for our research work. In section 4, we will detail and analyse the change management mechanisms of Protégé [4], which is one of the most popular ontology editing and dissemination tools currently available. What we proposed here is to enable the current change management mechanism in the Protégé system to make changes performed on the ontologies semantically traceable. Section 5 will describe a plan for building the diverting system mentioned above to enable external applications to continue to run in spite of any possible changes to the underlying ontologies. The report will be concluded in the section 6.

2 Motivations

For the purpose of efficiently reusing and managing existing ontologies, our research is motivated under this scenario: during the update process of an ontology (O_{old}*: old ontologies;* O_{new}: *new ontologies*) which is underlying an application (*A*), some of the services (*S*) that application *A* provides could become invalid because the service calls to the old version ontology O_{old} will not be available after changing to ontology O_{new}. As a result, users will not get the continuous, unchanged and consistent services *S* as before.

We observed that most of existing works which investigate ideas, methods and solutions for ontology versioning and evolution are limited to looking at the relationship and interoperability between the various ontology versions. Not much attention has been given to how ontology change might affect depending applications and services.

We believe that ontology versioning and evolution solutions should be applied to enable existing applications to adapt to newly updated ontologies. We also believe that in some situations (e.g. very big ontologies or when applications needs to be informed of any ontology changes) it might be useful to have a more focused ontology versioning control, where the system can keep an eye on possible changes to only certain places in an ontology, rather than to the entire ontology. This could help providing a more personalised versioning management service. To enable this service, the system will need to identify the important parts ("hot spots") of an ontology with respect to specific tasks or applications (See Figure 1).

We could identify the "hot spots" in an ontology by looking at how application (*A*) queries it. We intend to build a middle layer between the applications and the underlying ontology. This middle layer could monitor and detect any changes performed on the "hot spots" specific for an application, and re-divert queries accordingly, or even inform the application of those changes (e.g. by embedding some information in the returned messages).

Ontology winnowing techniques will be used to identify the "hot spots" of an ontology which are important for a specific application. This framework is an application-specific ontology management approach.

We argue that as the life of ontologies keeps continuing important parts ("hot spots") of the ontologies related to the specific applications should be kept evolving in the fine-grained and well-modularized manner so as to keep the whole ontologies in a "healthy" status.

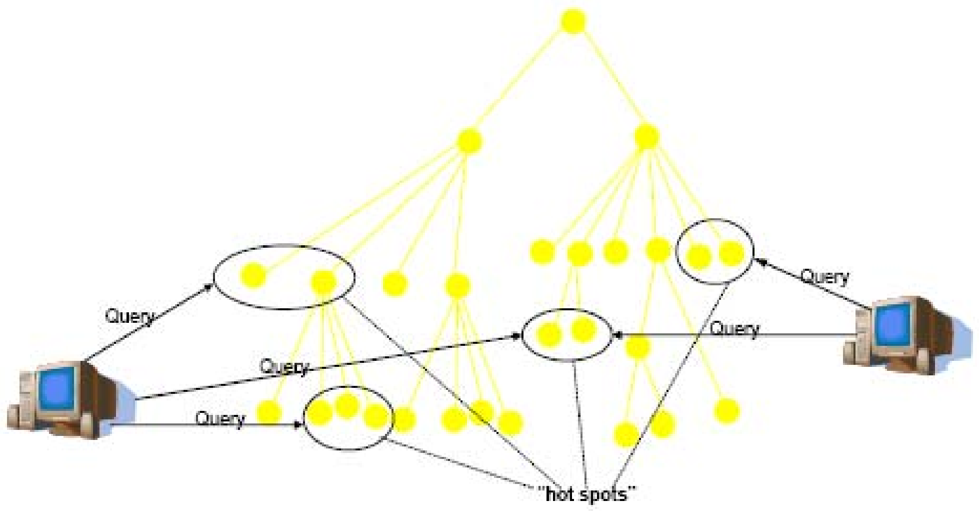

Figure 1 Identifying "hot spots" by monitoring applications' queries to the underlying ontology

3 Technical Background

3.1 Ontology Management and Knowledge Technologies

The reason ontologies are becoming popular is largely due to what they promise: *A shared and common understanding of a domain that can be communicated between people and application systems*. As such, the effective management and maintenance of ontologies offer an opportunity to enable ontologies to better represent human beings' updated conceptualization of the world, to improve the further use of the ontologies as well as to discover the potentials of the existing ontologies.

Ontology management is a whole set of methods, methodologies and techniques which are designed to effectively use/reuse the various (existing) ontologies from the different sources for the different tasks [9]. This report only focuses on the technologies targeted to server our goals described above (Section 1.2). These technologies are identified as ontology versioning and evolution. Meanwhile, the technologies related to ontology versioning and evolution will be depicted as well. For example the schema versioning and evolution in database and change management in medical field inspired ontology researchers to develop technologies to cope with versioning and evolution issues of ontologies.

Ontology is a valuable and explicit representation of knowledge. Research in technologies for accessing and managing knowledge could assist us to understand the way to cope with the issues in analogous areas within ontology research. Advanced Knowledge Technologies Interdisciplinary Research Collaborations (AKT IRC) project aims to develop and extend a range of technologies providing integrated methods and services for the capture, modelling, publishing, reuse and management of knowledge. Our research falls into this context.

Knowledge maintenance is the prerequisite of reuse, which is used to ensure the continuing usability of knowledge models and the relevant artefacts. It is a huge challenge to keep the knowledge exploited within an application functional all the time. Because knowledge is not static, the support of versioning and evolution will be essential and necessary to knowledge maintenance. This may involve deep analysis of the knowledge itself. Verifying and validating the knowledge content to secure its longevity in the application will be the core task for versioning and evolution, also the heart of knowledge maintenance.

3.2 Schema versioning and Evolution in Databases

The research in ontology versioning and evolution has borrowed many ideas from schema versioning and evolution in database research. In addition, ontology versioning and evolution are still in their early stages. Thus, it might be beneficial to look at the analogous solutions proposed for the database schema versioning and evolution first before tackling the same problem in ontologies.

Database systems are rarely stable following the initial implementation. Modifying the database schema is a common but often troublesome occurrence in database administration. Schema versioning and evolution arose in the context of long-lived database applications, where stored data were considered worth surviving changes in the database schema [10]. According to the definitions given in a consensual glossary [11], schema evolution allows the databases to modify the schema without the loss of extant data; furthermore, schema versioning enables the databases to query all of the data by means of any schema version according to the preference of the user or the application. Schema evolution can be considered as a special case of schema versioning [10] where only the current schema version is maintained based on the definitions and arguments in [12].

Since the introduction of schema change facilities in the database systems, two fundamental problems were solved: **semantics of change** and **change propagation** [10]. **Semantic of changes** involves the checking and maintenance of schema consistency after changes. Most existing approaches that address the semantics of change use *variants* to define the consistency of a schema, and definite *rules* that check the consistency of those *variants* after each change performed on the schema. For example, ORION [13] and O_2 [14] systems use this approach to support schema versioning and evolution. In addition, a set of *axioms* are used to formalise the dynamics of schema evolution which is the actual management of schema changes in a system in operation. **Change propagation** involves the consistency of extant data with the modified schema. In most cases [13, 14], simple default mechanisms can be used or user-supplied conversion functions must be defined for non-trivial extant object updates. [12] provides an excellent survey on the issues about schema versioning and evolution in the database systems, including modelling, architecture, and query language issues.

In [32], the author discussed that the problem of versioning and evolution in ontologies is significantly different with that in relational databases. Those differences stems from the different models and different usage paradigms. For example, ontologies allow the specification of multiple-inheritance.

3.3 Change Management in Controlled Medical Terminology

Controlled medical terminologies are also called health vocabularies [15], structured clinical vocabularies [16], controlled medical vocabularies [17], and reference terminologies [18]. A controlled medical terminology is a fundamental requirement in a range of medical informatics applications such as hospital departmental systems, patient record systems, expert systems, and medical literature databases [19], also it share the requirements for a number of important tasks, for example, searching for a term, decoding and encoding, retrieval of information about concept meaning, translation to other coding schemes and management of additions, modifications, and obsolescence [20].

Natural language changes over time, and the language for the medicine is no exception as well. For example, we now know of diseases that were unknown of in the past, and as we learn more about known diseases, we update them with new names that reflect our increasing and enhanced understanding for those diseases. A terminology that is useful today will not be useful tomorrow if it is not updated as the time being. Therefore, changes in the controlled medical terminologies are quite common.

The traditional approach in management of changes/updates of the controlled medical terminologies is by maintaining a long-list historical file of the vocabulary changes so that a given code on a given datum can be determined [17]. This method was appealing because of its simplicity and ability to support retrieval based on the codes which is used for coding the patient data. However, the retrieval based on the codes is of limited usefulness for clinical information applications where concept-based retrieval is required which refers to that each code is associated with a particular meaning. In concept-based vocabularies, changing the preferred name of a concept is permissible as long as the original meaning is preserved; if the meaning changed, then a new concept with the relevant new code is created. The National Library of Medicine's Unified Medical Language System (UMLS) takes a concept-based approach [21].

The researchers in medical fields has recognized that the researches in the knowledge representation field focus on concepts rather than on terms, and the hierarchies were typically formal taxonomic structure to which the algorithms for classification could be applied. Cimino in [19] argues for following a knowledge-based approach to manage medical terminology; Rector and colleagues populated an implementation of the description logic GRAIL with medical terms [22]. Gradually, the medical informatics community has recognized the desirability of approaches more formal than those used in MeSH, SNOMED, and ICD-9-CM.

3.4 Ontology Versioning and Evolution

After ontologies have been deployed, they inevitably have to change if they are to remain useful. Post-deployment of ontology will make new requirements emerge so as to call the existing requirements to change. These changes are not simply concerned with repairing faults omitted during the ontology construction phase. The majority of changes are a consequence of new requirements that are generated in response to changing environments and human beings' new understanding of the conceptualization of the world [31].

Research on ontology versioning and evolution focuses on the issues of how ontologies deal with the internal and external changing environments. According to [32], ontology versioning and evolution is defined as "*the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology*". This ability includes methods to distinguish and recognize versions, specifications of relationships between versions, update and change procedures for ontologies, and access mechanisms that combine versions of an ontology and the corresponding data.

In the last decade, majority of active researches in the area of ontology engineering focus on ontology construction. Coping with the changes and providing maintenance facilities were largely neglected. The existing commonly agreed methodologies and guidelines for ontology evolution are still blank [33]. Although evolution of ontologies as a requirement for successful application of ontologies is recognised widely, there are few approaches investigating the problems of change in ontologies [34].

Recently, researchers have proposed several directions to support ontology versioning and evolution. One of them is based on using explicit log of changes. Oliver et al. in [20] discussed the kind of changes that occur in medical terminologies and proposed the CONCORDIA concept model to deal with these changes. They specified a set of changes and described mechanisms for synchronizing different versions based on change logs. The main characteristics of CONCORDIA are that all concepts have a permanent unique ID; that concepts will be given a *retired* status instead of being really physically deleted; that based on the *retired* status special links are maintained to track the retired parents and children relationships. However, this approach could not be moved to the Semantic Web since there are no possibilities to control the whole process. Stojanovic et al. in [33] introduced the notion of *evolution strategies*, which allows developers to specify complex effects of change, in the context of KAON[6] suite of tools or ontology management. KAON uses logs of changes as versioning information. Another research work based on the log of changes comes from Ognyanov et al. [35]. RDF statements are the pieces of knowledge on which they operated. They propose a method for tracking changes in RDF repositories. However, the granularity of RDF statements is much lower than the granularity of class and property changes.

While there are many ontology tools/systems providing logs of changes between various versions to support ontology versioning and evolution processes, there is no interaction or sharing of information among these tools/systems. Having a general framework for ontology versioning and evolution that allows tools supporting different evolution tasks to share change information and leverage change information obtained by other tools, will make ontology versioning and evolution process much more efficient [38].

Another approach in supporting ontology versioning and evolution process is based on the comparison of the different variants/versions of the same ontology. Protégé ontology development system [36] and OntoView [37] are two examples of this approach. OntoView is an ontology versioning system which is based on the comparison of two ontology versions in order to detect change. It performs a pair-wise comparison of the sets of RDF statements that forms the old and new

[6] The KAlsruhe ONtology and Semantic Web tool suite (KAON): http://kaon.semanticweb.org/

version of the class- and property-definitions in an ontology. Thus, changes in the syntax and specific representation of RDF are ignored. Even though there are many ways to transfer one ontology into a new version, this system generates only one solution based on sets of heuristics [34]. The analysis of the support of ontology versioning and evolution facilities in Protégé will be detailed in next section.

4 Work Completed

During the last nine months, a range of activities has been undertaken under the banner and support of the Advanced Knowledge Technologies Interdisciplinary Research Collaborations.

The following activities and events have been attended:

- Regular attendance of AKT business and research meeting
- First Advanced Knowledge Technologies Semantic Web Services Workshop (AKT-SWS 2004, Milton Keynes)
- AKT Workshops (Southampton, Milton Keynes)
- AKT Doctoral Colloquium (Milton Keynes)
- Postgraduate Research Conference in Electronics, Photonics, Communications and Networks, and Computing Science 2005 (PREP 2005, Lancaster)

Along with the events I attended, I presented a poster titled '*Change Management: The Core task of Ontology Versioning and Evolution*' on [23]; in addition, I have a paper titled '*Ontology Change Management in Protégé*' was published in AKT Doctoral Colloquium [24].

The remainder of this section gives an overview of the work undertaken. The research work described below relies heavily on the literature review stated in section 3.

4.1 Analysis of Change Facilities in Protégé

Ontology schemas tend to change and evolve over time to meet new requirements. This change may invalidate dependent applications if there is no dynamic adaptation to the changes performed on the underlying ontologies. Unfortunately, appropriate tools for managing ontology evolution efficiently are still missing [28]. To capture the changes performed on the ontologies, knowledge engineers locate and identify the changes between versions of an ontology mainly by comparison [25]. Once conceptual relations between various versions are built up, it becomes possible to re-interpret the data and knowledge under the different versions so that incompatibility caused by the changes made to the ontologies could be resolved semantically [26].

Our research work selected Protégé as a test-bed to run our experiments.

Protégé evolved over the years from the role to reduce the knowledge-acquisition bottleneck in constructing knowledge bases to an extensible, customisable, and more general-purpose toolset for constructing knowledge bases (KBs) and developing KB-based applications [27]. As ontology development became a more ubiquitous and collaborative process, to maintain Protégé's robustness in future ontology-related applications, support for ontology management becomes essential and necessary. This support should allow the ontology developers to examine and understand the changes and their rational, and enable any dependent applications to remain up to date and compatible with the latest version of the ontology.

4.1.1 Undo/Redo

Like many other development tools, Protégé has an undo/redo function which allows users to re-track their previous steps. In addition, Protégé has a *Command History* menu item, which is used to record each of the change actions the user have made to the ontology. The *history* records are timely ordered according to when the change took place, and will be deleted once the Protégé project is closed.

4.1.2 Ontology Version Archiving

This function provides an easy interface for users to manage different versions of an ontology. It allows users to save the current version of an ontology and add any comments to it. The saved versions will be put into separate directories with different time stamps. Protégé users can revert to any saved version using *Revert to a Previous Version*.

4.1.3 PROMPT

PROMP is a plug-in suite for Protégé used to manage multiple ontologies. It has four main functions:
- **Compare** the current ontology to a different version of the same ontology;
- **Move** frames between the current including project and one of the included projects;
- **Merge** two ontologies and added the resulting merged ontology to the current project;
- **Extract** a portion of another ontology and add it to the current project.

The compare function is the most relevant to the subject of our research. Within compare, there are two kinds of views; Tree View and Table View. In the Tree View, ontology structure is displayed and the changes are highlighted. This view aids the ontology engineers to accept/reject the changes made to the ontology. The Table View enables users to save the change record as an output file (text file) for further use.

By enabling the "journaling" preference in the system (Project|Configure…|Options), some information about the changes made to the ontology (e.g. author, time, item changed) will be saved to a file with the extension ‘.pjrn’. PROMPT is able to read such files and retrieve some information about those changes.

According to the analysis of the current change facilities within Protégé, a set of requirements for a more Semantic Web friendly change management in Protégé is given out in the following section.

4.2 Semantically Represented Ontology Change

One of the important functionalities related to change management is logging. Logging changes is necessary to track any modification applied to the ontology. Protégé offers some limited support to ontology change logging as described earlier. Even though this logging mechanism is useful for Protégé services such as PROMPT, it can not be easily used by external services as it is not written in a machine understandable format. This is due to the reliance on Protégé's API call names in the log file, which can not be interpreted by non- Protégé programmers. Our aim is to be able to log ontology changes in a format that can be parsed and understood by general ontology-based application to allow such applications to track those changes and perhaps update their communications with the ontology accordingly.

4.2.1 Semantic Logging

A semantic representation of ontology change logs may enable other developers and tools to process and understand the evolution history of an ontology [29]. The type of changes that ontology engineers tend to make is highlighted in [28] and [29]. Other work investigated how best to index changes in medical terminologies to ensure their traceability [30] [17]

In our approach, we envisage a logging file to include the following aspects:

- Creating a system log ontology of change for each project;
- Populating the log ontology with instances to represent the details of any changes applied to the ontology;
- Encouraging ontology developers to write change rational and comments;
- Providing a facility to reconstruct the actions that led to the current state of the ontology;
- Providing access to the logging ontology to external tools and applications.

5 Future Work

Personal potential research interests focused on applying ontology versioning and evolution process semantically in the end-user of the applications/services and managing this process in a semantic fashion.

Future research activities are undoubtedly going to include the continuation of the existing work undertaken to enhance the current change facilities of ontology versioning and evolution within Protégé framework, including three main perspective works:

- Enabling system log ontology of change for each project;
- Enabling to re-construct the actions that led to the current state of the ontology;
- Enabling the external tools and applications to access the logging ontology

Most of the existing works which investigate the ideas, methods and solutions about ontology versioning and evolution are limited in the range of the relationship and interoperability between the various ontology versions. None of the works have touched the application spectrum where ontology versioning and evolution mechanisms are applied to secure the continuous services as the underlying ontologies change. Our research work will challenge this problem.

In the near-term, a service-diverting infrastructure will be built to demonstrate the application of ontology versioning and evolution for the end-user of applications/services within a specific domain. The initial framework for this system is presented in Figure 2.

In current implementation, during the ontology upgrade process, when users request some services from the applications as usual, the responses from the applications may be unavailable, or the results from the applications may be inaccurate. When applications/services query the changed ontologies, the new versions of the concepts in the changed ontologies might not recognized by the applications/services. With the addition of the middle diverting layer, it will make the existing applications/services adapt seamlessly to the underlying updated ontologies so as to provide continuous and unchanged services as before by automatically diverting services' queries which use the old versions of the ontologies to use new available versions.

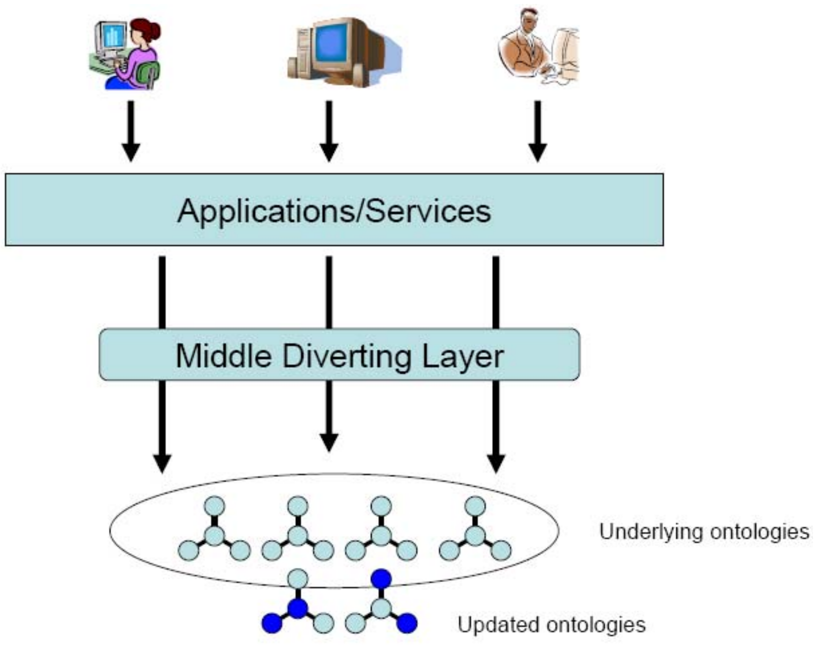

Figure 2 The framework of initial services diverting system

Central to this diverting system are two principal tasks. One is to semantically manage changed ontologies, which will include the tasks of identifying ontology changes, and keeping record of ontology changes in a semantic fashion so as to enable the changes traceable. The other task is to create a dynamic diverting mechanism for the services by mapping the concepts in the old and new ontologies respectively. The mapping sources in the second task are coming from the semantic representation of the ontology changes in the first task.

In choosing a domain to work on, taking the fact into consideration that the changes performed on the ontologies should be kept since their origins as intact and consistent as possible so as to be convenient to be processed and investigated, Conceptual Reference Model (CRM)[7] is the most obvious as it provides the detailed documentation about the changes as well as it provides different familiar and accessible encoding formats, such as XML, RDF, HTML and so on, for the different popular applications. In addition, the ontologies in the medical field may also be the choice for the research domain because of the personal background in medicine.

So, given the prescribed the infrastructure of the framework and the descriptions of the core technologies, it is expected to produce a working system demonstrating the application of ontology versioning and evolution for the end-users of the applications/services.

6 Acknowledgement

I would like to thank my supervisors and colleagues for allowing me to contribute to a variety of interesting and stimulating work.

Part of this work has been supported under the Advanced Knowledge Technologies Interdisciplinary Research Collaboration (AKT IRC), which is sponsored by the UK Engineering and physical Sciences Research Council under grant number GR/N15764/01.

7 References

[1] Denny, M., *Ontology Tools Survey, Revisited.* July, 2004. Online: [URL: http://www.xml.com/pub/a/2004/07/14/onto.html, Accessed in June 2005]

[2] Popkin, J., *Top 10 Strategic Technologies for 2005.* Gartner Symposium ITxpo 2004, Orlando, USA, October 29, 2004

[3] Denny, M., *Ontology Building: A Survey of Editing Tools.* November, 2002. Online: [URL: http://www.xml.com/pub/a/2002/11/06/ontologies.html, Accessed in June 2005]

[4] Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., Tu, S. W., *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development,* 2002

[5] World Wide Web Consortium, *Semantic Web Activity Statement,* 2001. Online: [URL: http://www.w3.org/2001/sw/Activity, Accessed in June 2005]

[6] Berners-Lee, T., Hendler, J. and Lassila, O., *The Semantic Web,* Scientific American, May 2001.

[7] Uschold, M., *Where are the Semantics in the Semantic Web?,* AI Magazine, 2003, 24 (3), pp25-36

[8] Fensel, D., Bussler, C., Ding, Y., Kartseva, V., Klein, M., Korotkiy, M., Omelayenko, B., and Siebe, R., *Semantic Web Application Areas,* Oracle White Paper, March 2nd, 2002. Online: [URL: http://whitepapers.zdnet.co.uk/0,39025945,60120659p-39000539q,00.htm, Accessed in June 2005]

[9] Davies, J., Fensel, D., and Harmelen, F. V., ed., *Towards The Semantic Web: Ontology-driven*

[7] The CIDOC CRM provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation. [URL: http://zeus.ics.forth.gr/cidoc/index.html]

Knowledge Management. John Wiley & Sons, Ltd, England, 2003

[10] Franconi, E., Grandi, F., and Mandreoli, F., *A Semantic Approach for Schema Evolution and Versioning in Object-Oriented Databases*, In: Proceeding of the Sixth International Conference on Rules and Objects in Databases (DOOD 2002), 2002[10]

[11] Jensen, C. S., Clifford, J., Gadia, S. K., Hayes, P., and Jajodia, S., et al, *The Consensus Glossary of Temporal Database Concepts*, February 1998 Version. In: Etzion, O., Jajodia, S., and Sripada, S., editors, *Temporal Databases - Research and Practice*, Springer-Verlag, 1998, pp 367-405[11]

[12] Roddick, J. F., *A Survey of Schema Versioning Issues for Database Systems*, Information and Software Technology, 37 (7), 1995, pp383-393[12]

[13] Banerjee, J., Kim, W., Kim, H. J., and Korth, H. F., *Semantics and Implementation of Schema Evolution in Object-Oriented Databases*. In: Proceedings of the ACM-SIGMOD Annual Conference, May 1987, pp 311-322[13]

[14] Ferrandina, F., Meyer, T., Zicari, R., Ferran, R., and Madec, J., *Schema and Database Evolution in the O2 Object Database System*, In: Proceedings of International Conference on Very Large Databases (VLDB), September 1995, pp170-181[14]

[15] Humphreys, B. L., Hole, W. T., McCray, A. T. and Fitzmaurice, J. M., *Planned NLM/AHCPR large-scale vocabulary test: using UMLS technology to determine the extent to which controlled vocabularies cover terminology needed for health care and public health*, Journal of the American Medical Informatics Association 3, 1996, pp 281–287[15]

[16] Kannry, J. L., Lawrence, W., Shifman, M., Silverstein, S. and Miller, P. L., *Portability issues for a structured clinical vocabulary: mapping from Yale to the Columbia Medical Entities Dictionary,* Journal of the American Medical Informatics Association, 3, 1996, pp 66–79[16]

[17] Cimino, J. J., *Formal descriptions and adaptive mechanisms for changes in controlled medical vocabularies*, Methods of Information in Medicine 35, 1996, pp 202–210[17]

[18] Spackman, K. A., Campbell, K. E. and Cote, R. A., *SNOMED-RT: A reference terminology for health care*, In: Masys, D. R., ed., Proceedings for the 1997 Annual AMIA Fall Symposium (Hanley & Belfus, Washington, DC, 1997) pp 640–644[18]

[19] Cimino, J. J., Clayton, P. D., Hripcsak, G., and Johnson, S. B., *Knowledge-based Approaches to the Maintenance of a Large Controlled Medical Terminology,* Journal of the American Medical Informatics Association, 1 (1), 1994, pp 35-50[19]

[20] Vancouver, B. C., Oliver, D. E., Shahar, Y., Shorliffe, E. H. and Musen, M. A., *Representation of Change in Controlled Medical Terminologies*. Artificial Intelligence in Medicine 15, 1999, pp 53-76[20]

[21] Tuttle, M. S., Olson N. E., Campbell, K. E., Sherertz, D. D., Nelson, S. J., Cole, W. G., *Formal Properties of the Metathesaurus,* In: Ozbolt, J. G., ed, Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care, New York, McGraw-Hill, 1994, pp 145-149[21]

[22] Rector, A., Bechhofer, S., Goble, C., et al, *The GRAIL Concept Modelling Language for Medical Terminology,* Artificial Intelligence in Medicine, 9 (2) 1997, pp 139-171[22]

[23] Liang, Y., Alani, H., Shadbolt, N.R., *Change Management: The Core task of Ontology Versioning and Evolution.* Postgraduate Research Conference in Electronics, Photonics, Communications and Networks, and Computing Science 2005 (PREP 2005), Lancaster, 30.03.2005-01.04.2005[23]

[24] Liang, Y., Alani, H., Shadbolt, N.R., *Ontology Change Management in Protégé*. In: Proceeding of Advanced Knowledge Technologies Doctoral Colloquium, Milton Keynes, United Kingdom,

06. 2005[24]

[25] Noy, N.F., Musen, M.A., *Ontology Versioning in an Ontology Management Framework.* IEEE Intelligent Systems, Vol. 19, No. 4, July – August, 2004[25]

[26] Klein, M., Fensel, D., *Ontology Versioning on the Semantic Web*, Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA, July 30 -- Aug. 1, 2001[26]

[27] Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubezy, M., Eriksson, H., Noy, N.F., Tu, S.W., *The evolution of Protégé: an environment for knowledge-based systems development*. In: International Journal of Human-Computer Studies, 2003, 58 (1) pp89-123[27]

[28] Sure, Y., *On-To-Knowledge -- Ontology based Knowledge Management Tools and their Application*, In: German Journal Kuenstliche Intelligenz, Special Issue on Knowledge Management (1/02), 2002[28]

[29] Maedche, A., Motik, B., Stojanovic, L., Studer, R., Volz, R., *Managing Multiple Ontologies and Ontology Evolution in Ontologging*, Intelligent Information Processing 2002, pp51-63[29]

[30] Oliver, D. E., Shahar, Y., *Change Management of Shared and Local Health-Care Terminologies,* IMIA Working Group 6 Meeting on Health Concept Representation and Natural Language Processing, AZ, U.S.A, 1999[30]

[31] Klein, M., and Fensel, D., *Ontology Versioning on the Semantic Web,* In: the Proceedings of International Semantic Web Working Symposium (SWWS), Stanford University, California, U.S.A, 2001[31]

[32] Noy, N. F., and Klein, M., *Ontology Evolution: Not the Same as Schema Evolution.* Knowledge and Information Systems, 5, 2003[32]

[33] Stojanovic, L., et al, *User-Driven Ontology Evolution Management.* In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Ontologies ad the Semantic Web, 2002, pp 285-300[33]

[34] Stojanovic, L., et al, *Ontology Evolution as Reconfiguration-Design Problem Solving,* In: International Conference on Knowledge Capture, Sanibel Island, Florida, U.S.A., 2003[34]

[35] Ognyanov, D., and Kiryakov, A., *Tracking Changes in RDF(S) Repositories,* In: 13th International Conference on Knowledge Engineering and Management, EKAW 2002, Spain, 2002[35]

[36] Noy, N. F., Kunnatur, S., Klein, M., and Musen, M. A., *Tracking Changes During Ontology Evolution.* 3rd International Semantic Web Conference (ISWC2004), 7-11 November 2004, Hiroshima, Japan[36]

[37] Klein, M., Kiryakov, A., Ognyanov, D., and Fensel, D., *Ontology Versioning and Change Detection on the Web,* In: 13th International Conference on Knowledge Engineering and Management (EKAW 2002), Siguenza, Spain, 2002[37]

[38] Klein, M., and Noy. N. F., *A Component-Based Framework for Ontology Evolution,* Technical Report IR-504, Department of Computer Science, Vrijie Universiteit Amsterdam, March 2003[38]