

NRC Publications Archive Archives des publications du CNRC

Constructing Knowledge Bases for E-Learning Using Protégé 2000 and Web Services

Hogeboom, M.; Lin, F.; Esmahi, L.; Yang, Chunsheng

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version.
/ La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

*Proceedings of The International Conference on Advanced Information
Networking and Applications (AINA 2005), 2005*

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
<https://nrc-publications.canada.ca/eng/view/object/?id=60483a83-8c9a-4028-b8b3-6629b606cfc>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=60483a83-8c9a-4028-b8b3-6629b606cfcb>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the
first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez
la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous
n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



National Research
Council Canada

Conseil national
de recherches Canada

Institute for
Information Technology

Institut de technologie
de l'information

NRC - CNRC

Constructing Knowledge Bases for E-Learning Using Protégé 2000 and Web Services *

Hogeboom, M., Lin, F., Esmahi, L., and Yang, C.
March 2005

* published in The Proceedings of the International Conference on Advanced Information Networking and Applications (AINA 2005), Taiwan. March 28-30, 2005. NRC 47441.

Copyright 2005 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Constructing Knowledge Bases for E-Learning Using Protege 2000 and Web Services

Mike Hogeboom¹, Fuhua Lin¹, Larbi Esmahi¹, and Chunsheng Yang²

¹Athabasca University, Canada

mikehogeboom@hotmail.com

{oscar|larbie}@athabascau.ca

²National Research Council, Canada

Chunsheng.Yang@nrc-cnrc.gc.ca

Abstract

This paper presents an approach to designing and developing knowledge bases in e-Learning systems. We explore the use of Protégé 2000 as a knowledge editor for course material, with the addition of Web Service interfaces on top of it to facilitate retrieval of the content. Protégé 2000 provides an extensible infrastructure and allows the easy construction of domain ontologies, customized data entry forms, and provides an API that can easily be extended by Web Services for the purpose of dynamic course material creation. The use of ontology and Web Services makes the knowledge bases for e-Learning sharable, reusable, and interoperable with other technologies, such as .Net, which have standard Web Service implementations.

1. Introduction

Currently there is a lack of knowledge bases (KBs) that can be accessed in a language and platform independent manner to support e-Learning. Being able to connect to a remote or local KB will allow the instructors or their intelligent software agents to create new courses (or course material for existing courses) from a central repository that is kept up to date with relevant information, and it has the potential to allow students to connect to the same repository to perform further research or view the most current version of the content.

Being able to access a central KB and search content on certain criteria would reduce the time to find relevant information on the particular knowledge domains stored in the model. Currently information has to be searched through the Web, libraries (print material), or some proprietary search technologies. Developing a KB with open standard accessibility will

enable adaptive course generation and facilitate delivery of course material.

2. Literature Review

T. Gruber's research on Portable Ontology Specifications deals with defining a common vocabulary in which shared knowledge is represented [1]. P. Saini has performed some research on Deriving Ontology-based Metadata for e-Learning from the ACM Computing Curricula [2].

Other research that is being performed for e-Learning knowledge bases deals with either designing and implementing an ontology within Protégé 2000 [3] or another data storage technology, or defining standards for data structures and queries of a domain model. Examples of such research include Chen's [4] research on "Communication Content Ontology for Learner Model Agent in Multi-agent Architecture" and Read's [5] research on "Incorporating interoperability into a distributed e-Learning system". Our research topic of coupling Web Services [6] to a Protégé 2000 knowledge base in particular appears to be unique.

According to Wiley, "Learning objects are elements of a new type of computer-based instruction grounded in the object-oriented paradigm of computer science" [7]. LOs are sometimes defined as being educational resources that can be employed in technology-supported learning. With appropriate metadata descriptions, they can be modular units that can be assembled together to form lessons and courses. [8]. The e-Learning system supports learning objects in a fine grained manner, meaning that the components of a publication are broken into smaller more concise pieces or "chunks" of information that can be retrieved individually, as opposed to retrieving a whole publication or even a whole chapter of a publication.

3. System Architecture

In our proposed approach, an e-Learning system mainly consists of a Protégé knowledge base, five Web Services that facilitate search and retrieval from the knowledge base, and a client that calls the Web Services and performs other functions, such as publishing the content to HTML. Figure 1 is a high level representation of the system. An e-Learning client communicates with the e-Learning server using SOAP messages over HTTP. The returned messages are stored in a client side session that can be saved for later retrieval, or published using XSLT to static HTML pages for use as course material.

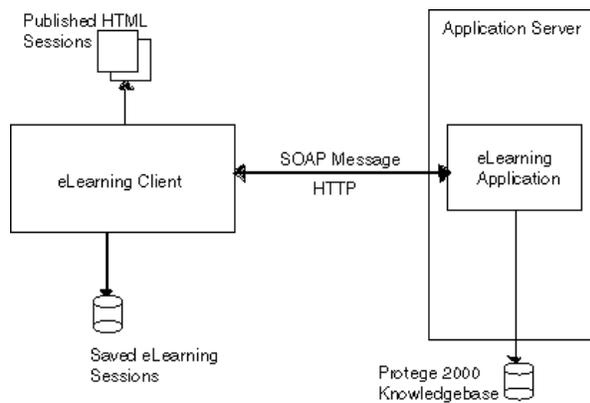


Figure 1: e-Learning Architecture.

4. Design of Protégé e-Learning Knowledge Base

Figure 2 shows the design of the Protégé e-Learning knowledge base in the form of a class diagram. From this diagram we can see that the “Content” is the main component of the knowledge base, and that it has a “Publication” which belongs to a certain “Subject Area”, and that each “Publication” has a different “Origin” (or author), and that a “Publication” can be from different sources, like “Book” or “Article”.

Separating a publication into multiple pieces facilitates using the Protégé search API to its maximum potential (and if a DB back-end was used to an even greater extent) and makes searches more powerful, precise, and content management more manageable once retrieved from the knowledge base. In the e-Learning KB, each learning object or content has a corresponding publication, which belongs to a specific “Subject Area”. Each “Publication” has an “Origin” (or multiple “Origins”), which is an abstract type and must be one of either “Person” or “Organization”. If this ontology were to be extended in the future other

types of “Origins” could easily be added to suite the needs of the implementation. Since “Publication” is an abstract type, it must have a concrete child implementation to which the “Content” belongs. The concrete child classes are shown in the figure below along with the rest of the classes involved in the e-Learning Protégé 2000 knowledge base ontology.

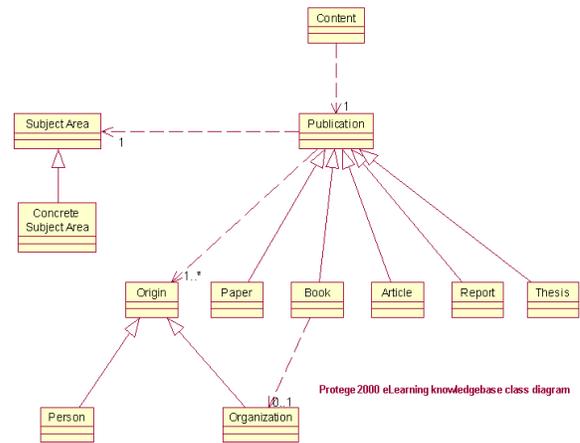


Figure 2: Class diagram for e-Learning Protége 2000 knowledge base

This design of the Protégé 2000 knowledge base was deemed to be the best solution for this particular problem. Other designs, such as having the “Content” as an abstract type and having child classes based on topic or publication type would have also worked but did not seem to be the best choice in this situation because searches on “Content” would have been slower and more complex if designs such as that were chosen because multiple queries would have been required due to the underlying structure of the Protégé 2000 data format and the available API.

Figure 3 shows how the Protégé 2000 classes relate to each other within the Protégé 2000 application. If compared to the class diagram of the ontology design above it can be seen how the classes relate to each other. This figure varies from the class diagram in that it shows the actual subject areas that were added to the knowledge base (e.g. Computer Science).

The following describes an example, Content class, of the individual classes within the e-Learning KB and comments on their slots and facets.

The content class is the key class in the e-Learning ontology. Instances of the content class represent individual pieces of content that belong to particular publications. Figure 4 depicts the content class as it is

represented in Protégé 2000. Each piece of content has an owning publication, which is an individual instance of a concrete publication class. Each content class can also potentially have any number of web graphics associated with it. The web graphic image itself is not located in the knowledge base, but rather the relative path to its location on the web server running the e-Learning application. Contents also have previous and following content associated with it. These are the contents that precede and follow the current piece of content. By using the previous and following content it is possible to retrieve an entire publication from the knowledge base to display it as a whole. Other textual slots in a content class include the text itself, the content title, section, subtitle, and keywords.

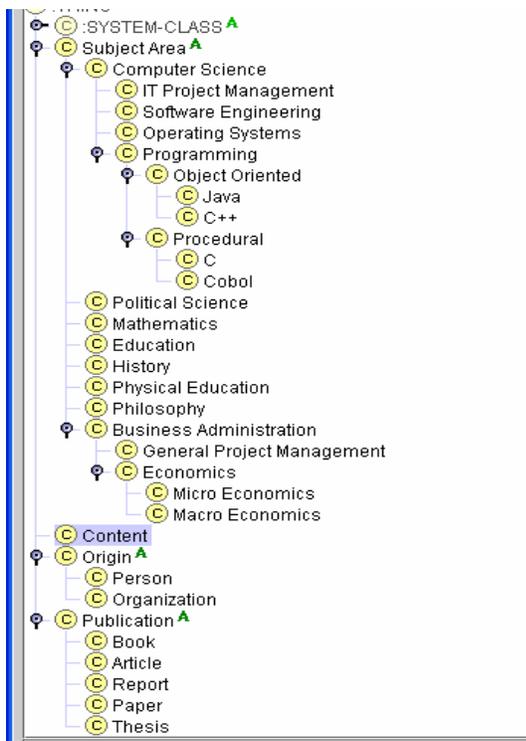


Figure 3: Protégé 2000 hierarchy of e-Learning knowledge base

5. e-Learning Server Design

The e-Learning server application is comprised of a J2EE application server containing the e-Learning application and the Protégé 2000 e-Learning knowledge base. When using the Java Web Services Developer Pack to generate services, it is necessary to create the endpoint interface and implementation of the class that is going to be exposed as a Web Service. In the case of the e-Learning project there are five pairs of interfaces and classes, they are:

- GetContentIF → GetContentImpl

- GetPublicationIF → GetPublicationImpl
- GetTopicsIF → GetTopicsImpl
- SearchContentIF → SearchContentImpl
- SearchPublicationsIF → SearchPublicationsImpl

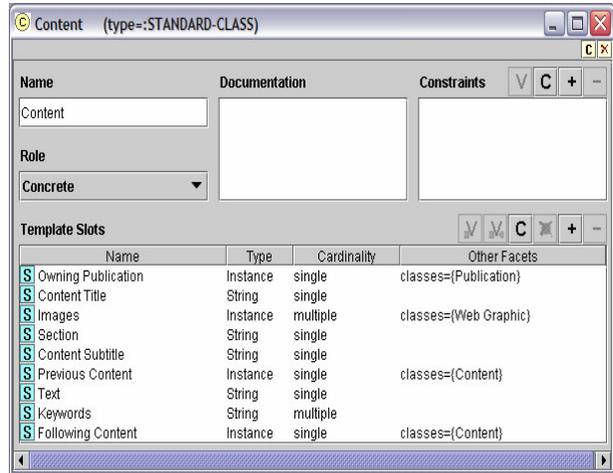


Figure 4: Protégé 2000 content class of e-Learning knowledge base.

The interfaces declare the methods that a remote client (the e-Learning client) can invoke on the service. The class implementations are the concrete realizations of the service and they are also the Data Access Objects (DAOs) that are used to query the Protégé 2000 e-Learning knowledge base for the requested information. During the build process, these interfaces and classes are used to generate the JAX-RPC service that is deployed.

6. e-Learning Client Design

The e-Learning client consists of a set of Java classes that capture user input from the command line and call client side business commands that invoke the calls to the e-Learning Web Services described Section 7 of this document.

The e-Learning client can be used to search for content, list topics, search for publications, save sessions, load previously saved sessions, view sessions, and publish sessions to static HTML files.

The following sections describe how the client is used from a user's perspective. The following is a list of actions that can be performed by the e-Learning client:

- Load a previously saved session
- Create a new session
- Save a session
- View contents of a session
- Add content to the session

- Publish a session to HTML
- Search the e-Learning Web Services for content based on certain search criteria
- Search the e-Learning Web Services for authors based on certain search criteria
- Traverse forwards and backwards through individual pieces of content contained within a publication once a piece of content is found (essentially you could read the entire publication in entirety if you so desired)
- View all the topics/subject areas stored in the e-Learning knowledge base by querying the e-Learning Web Services

When session is referred to when describing the e-Learning client, it refers to the current user interaction with the e-Learning client. Only one session can be active at a time. Those sessions not being used can be saved before a new session is created or are lost once a new session is made.

7. Implementation

We have developed a prototyping system that can be used to perform the intended task of adaptive course generation and delivery. The work involved included determining how to model the domain within Protégé 2000 to suit the needs of this particular problem, and determining how to access the contents of the knowledge base, once populated, and search and retrieve information from the knowledge base. Another aspect that was researched in the course of this project was how to tie all the pieces together through Web Services, which included developing the Web Services themselves, generating the corresponding WSDL files, and creating a Java client to access the Web Services through JAX-RPC.

During the research for this project it was necessary to determine whether the Protégé 2000 application was even flexible and open enough to extend the Protégé API with Web Services and whether it supported the functions required for this project to search a knowledge base. It was also necessary to determine what types of Web Services were most useful/appropriate for this project.

Currently there is no printed or Web-based publication on the Protégé API so it was necessary to review the JavaDocs (Java Documentation) and message boards for information on how to query the knowledge base. There is a User Guides for Protégé 2000 itself, on how to use it and create and design ontologies and KB, but the documentation for using the Protégé API consists of message boards and a few short articles with some example programs. It was

necessary to read through the JavaDocs and example programs/message boards to determine whether it was even feasible to connect a Web Service to the knowledge base and whether the required search criteria was even supported by the API. As of June of 2004, the JavaDocs did not contain a great deal of comments for the functions and even some classes, but fortunately, along with the message boards, example articles, and the fact that the class and function names were meaningful it was possible to determine how to use and extend the Protégé APIs.

It was proposed that the e-Learning application would utilize a Protégé 2000 KB which stored small pieces of contents (learning objects), as opposed to storing the document as a whole in the KB. The granularity (size/focus) of each piece of content would be determined by the person entering the particular piece of content. For instance a newspaper article may be short enough to be entered as one piece of content, whereas a book would be separated into many pieces of content to maximize search capability. This approach was decided upon because it maximizes the search ability of the application. Following is the detailed description of several main components implemented in our prototyping system.

Web Services

The Web Services used in the e-Learning project are RPC-style (Remote Procedure Call – style) Web Services which have the following characteristics:

- Procedure call
- Method signature
- Marshalling
- Tightly-coupled
- Point to point
- Synchronous
- Typically within Intranet

RPC-style Web Services were chosen for this project because this type of service should be used within an enterprise, when there is reliable and high bandwidth, for short running business process, and when there is a trusted environment. These properties suit the e-Learning project therefore RPC-style was chosen. Document-driven Web Services are more suited for long running business processes and between enterprises therefore it was not chosen for implementing this project.

The e-Learning project consists of five Web Services listed below.

- GetPublication
 - This Web Service is used to retrieve a summary of a publication stored in the e-Learning KB. It takes a Protégé class ID as input

- The URL for the corresponding WSDL file is <http://<BASE-URL>/elearning-jaxrpc/GetPublication?WSDL>
- GetTopics
- This Web Service is used to list all the topics (Subject Areas) that are contained within the e-Learning KB. It returns an array of topics and each topic in turn contains an array of child topics
- <http://<BASE-URL>/elearning-jaxrpc/GetTopics?WSDL>
- SearchContent
- This Web Service is used to search the e-Learning KB for pieces of content that match the required search criteria. Searches can be performed based on keywords in the body of the content, author name, and topic (subject area)
- <http://<BASE-URL>/elearning-jaxrpc/SearchContent?WSDL>
- GetContent
- This Web Service is used to retrieve an individual instance of content stored in the KB. It will also retrieve some information about the publication it belongs to and the previous and following pieces of content.
- <http://<BASE-URL>/elearning-jaxrpc/GetContent?WSDL>
- SearchPublications
- This Web Service is used for searching for publications based on a specified search criterion. Publications can be searched based on: topic (subject area), description, title, and author name.
- <http://<BASE-URL>/elearning-jaxrpc/SearchPublications?WSDL>

GetTopics WSDL

The Topic object consists of a title and an array of subtopics. Figure 5 shows the Topic class.

The Topic class was designed in such a way that a tree of topics/subject areas could be created that could be traversed for display purposes. When calling the GetTopics Web Service an array of Topics is returned that is traversed recursively to display the contents in a properly formatted fashion, as depicted in Figure 6.

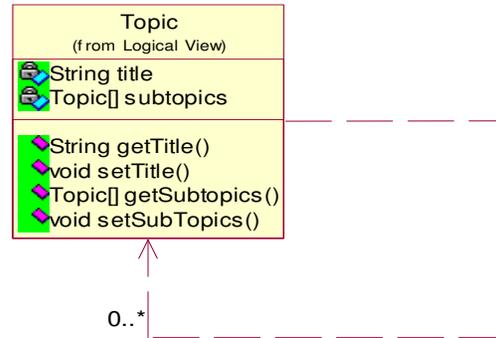


Figure 5: Class diagram of the Topic class

```

LISTING TOPICS, PLEASE BE PATIENT AS WE QUERY WEB SERVICE

Endpoint address = http://localhost:8080/elearning-jaxrpc/GetTopics
-Subject Area
  -Computer Science
    -IT Project Management
    -Software Engineering
    -Operating Systems
    -Programming
      -Object Oriented
      -Java
      -C++
      -Procedural
      -C
      -Cobol
  -Political Science
  -Mathematics
  -Education
  -History
  -Physical Education
  -Philosophy
  -Business Administration
    -General Project Management
    -Economics
      -Micro Economics
      -Macro Economics

Press <ENTER> to return to menu...
  
```

Figure 6: List Subject Areas.

An array of Topic objects was chosen for simplicity, the same functionality could have been achieved by using one of the several implementation classes of the java.util.Collection interface that are supported by JAX-RPC 1.3 [9].

e-Learning Client implementation

The e-Learning client was designed as a command line application with different input screens to capture and display data to/from the user. The e-Learning client consists of six packages containing the code developed from scratch for the e-Learning project.

8. Conclusions and future work

We have presented a methodology that using a design in which a publication as a whole is segregated into multiple pieces of content and stored in an appropriately designed Protégé 2000 knowledge base that retrieving content through Web Services is feasible, straightforward, and useful for the purposes of dynamic course creation. The design is flexible enough to allow different levels of content granularity to be

entered into the system depending on they type and informational qualities of the content/publication being entered.

This project also demonstrates the ease of use of the Java Web Services Developer Pack (JWS DP) combined with the Protégé 2000 Java API and how they can be extended to facilitate e-Learning and dynamic course creation. Furthermore, the use of a standard such as Web Services in the form of SOAP RPC calls allows remote clients written in different languages to connect to the e-Learning Web Services

Using a client such as the e-Learning Java client, it is fairly straight forward and intuitive how to go about retrieving content and adding it to a client side session, then publishing it to HTML or saving for later retrieval and use. This project, as a by-product, also shows how leveraging XSL technology can be beneficial in publishing the retrieved content because it is easy to change the look and appearance of published content by changing the XSL style sheets.

These Web Services could also potentially be used by students enrolled in a course or by employees of an organization to access content in real time as opposed to static published HTML, so there would only be one central knowledge base that is always current an up-to-date.

This project also demonstrates how distributed systems are useful in an e-Learning environment and how a central repository can be beneficial in keeping information consistent and up-to-date. Another advantage of a system such as this, is that by having a central repository that is queried for educational purposes, it would be possible to add a metrics layer to the system that could measure which information is being used and concentrate more resources on those subject areas.

The current implementation of the e-Learning application uses a command line interface to capture information from the user and display the results. In the future it would be advantageous to create a more robust and user friendly client using Java Swing technology. This way the input screens would be more visually pleasing and intuitive to use and the results can be displayed in a more useful way. Besides client user interface changes there are a number of other things that could be done to enhance the e-Learning project in the future. The Protégé knowledge base, for instance, could be upgraded to a relational database. Protégé 2000 supports any relational database that has a JDBC 1.0 compliant driver. Upgrading to a database for the storage format will improve search times which decrease as the knowledge base becomes larger. Upgrading the knowledge base format from Protégé

2000 1.9 to 2.0.1 would also be a beneficial enhancement to this project.

Another beneficial future enhancement that could be made is logging statistics on what data is being retrieved from the knowledge base and by whom. If this information was captured reports could be generated on which subject area is the most popular and resources could be focused on entering more data on that subject area into the knowledge base. A final suggestion for future enhancements would be adding more ways to query the knowledge base, either by extending the current Web Services or by creating new ones to support new queries that are desired and are deemed useful.

References

- [1]. Gruber, T. R. (1993), A Translation Approach to Portable Ontology Specifications, A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199-220.
- [2]. Saini, P., and Ronchetti, M. (2003). Deriving Ontology-based Metadata from E-Learning from the ACM Computing Curricula. University of Trento, Italy.
- [3]. Protégé Project. (2003). The Protégé Ontology Editor and Knowledge Acquisition System. Retrieved Dec 1, 2003 from <http://protege.stanford.edu/>
- [4]. Chen W., and Mizoguchi R. (1999). Communication Content Ontology for Learner Model Agent in Multi-agent Architecture. Retrieved May 6, 2004, from <http://www.ei.sanken.osaka-u.ac.jp/~aied99/a-papers/W-Chen.pdf>
- [5]. Read, T. and Verdejo, F. (2003): Incorporating interoperability into a distributed eLearning system. Retrieve May 6, 2004, from <http://sensei.lsi.uned.es/ea2c2/articulos/EM2003.pdf>
- [6]. Armstrong, E. (2003). The Java Web Services Tutorial., <http://java.sun.com/webservices/downloads/webservices/tutorial.html>
- [7]. Wiley, D. A. (2000). "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy," in D. A. Wiley, (ed.), *The Instructional Use of Learning Objects: Online Version*. Retrieved May 18, 2001, from the Web: , <http://reusability.org/read/chapters/wiley.doc>
- [8]. McGreal, R., Learning Objects: A Practical Definition, http://itdl.org/Journal/Sep_04/article02.htm
- [9]. Chowhury A., and Choudhary P. (2003). Working with JAX-RPC. Retrieved May 6, 2004, from <http://www.developer.com/java/ent/article.php/2109561>