AN ONLINE ONTOLOGY NAVIGATOR FOR WEB APPLICATIONS IN EDUCATION

Shireesha Tankashala, Fangfang Liu, Brian Horton and Shengru Tu
Computer Science Department, University of New Orleans, New Orleans, LA 70148, U.S.A.

Keywords: Ontology, Semantic Web, Computing Ontology, Information Retrieval, Training, METOC.

Abstract: We have developed a tree-view browser to support development of information retrieval applications that utilize ontologies. The purpose of this browser is to shield the complexity of an ontology from the users. We have also reported three ontology-based Web applications in education and training. The scope of these applications is to cover the daily activities of instructors and trainers. For example, a professor can create highly customized study guides using a bioinformatics ontology, a trainer can define the training needs for a specific task based on a network security ontology, and a domain expert can generate the Web forms for specialized training based on a METOC ontology. These applications were developed using our tree-view browser. The reuse of the browser component has significantly reduced efforts in the development.

1 INTRODUCTION

Recognizing that ontology is a core component of the semantic Web architecture (Berners-Lee, 2001), researchers and practitioners have invested significant efforts in establishment of ontologies in many areas. In early years, a methodology for ontology-based knowledge management was suggested in (Sure, 2002) which also reported an early case study of the methodology at a large enterprise. Happel envisioned a long list of adoption and utilization of ontologies in all the phases of software life cycle (Happel, 2006). A large number of exemplar ontologies in bioinformatics have demonstrated maturity and completeness (EBI, 2009). As to computer science education, Cassel has led the ambitious computing ontology project for many years (Cassel, 2007).

As Davies said: "the ontology itself is only as valuable as the applications that are developed to apply it to specific problems." (Davies, 2006) Observing the steady growth of the computing ontology at the "Computing Ontology Project" site (http://what.csc.villanova.edu/twiki/bin/view/Main/OntologyProject), we have realized that it is time to put more efforts in developing applications using ontologies in education. In this paper, we shall report three ontology-based applications in education and training scenarios. The scope of these applications covers the daily activities of instructors and trainers. For example, a professor can create highly customized study guides using a high-quality ontology, a trainer can define the initial training needs for a specific task based on the ontology of the field, and a domain expert can generate the Web forms for specialized training. In doing so, we have developed a tree-view browser that supports the "navigate, select, produce" three-step process which can effectively support information retrieval in various applications. As reported in (Ribaudo, 2009), "download of information" is the dominant activities in their university's virtual environment. We believe information retrieval will be the common activities in ontology applications.

The contribution of this paper is the effective approach that shields the complexity of ontology from the users and ease the development of ontology applications for education. The primarily targeted users are the instructors and trainers in the fields where some reasonable ontologies have been established. Advanced students and trainees can also join the user base. The diversified examples in this paper are to illustrate how effectively information can be retrieved with the help of ontologies. The focus of this paper is on the technical support to educational needs rather than plotting a new pedagogical technique.

The rest of this paper is organized as the following. In Section 2 related work is briefly surveyed. In Section 3, three applications are

Tankashala S., Liu F., Horton B. and Tu S. (2010).
AN ONLINE ONTOLOGY NAVIGATOR FOR WEB APPLICATIONS IN EDUCATION.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 319-324
DOI: 10.5220/0002802003190324

demonstrated followed by the description of our special tree-view browser for navigation in ontologies. The technical details of implementation are revealed in Section 4. Section 5 concludes.

2 RELATED WORK

The computing ontology project (Cassel, 2007; Davies, 2006) will have no doubt to take the central role in the curriculum development in computer science. A major emphasis of the project is to make the topics and the relationships between the topics available to instructors and curriculum developers. With such a long-term goal, defining relationships is very prudent. In (Cassel, 2007), only the basic, simple relationships, such as HasPart, Uses, Instance, and IsA (including their inverse) are considered. The has-prerequisite relationship was suggested in (Khan, 2007). Visualization was prudently planned for activities such as visualization of the union of the bodies of knowledge of the computing disciplines in (Cassel, 2007). Since our emphasis is on applications of ontologies rather than establishment of ontologies, we are more flexible in adding (overlapping) relationships to ontologies in the way described in Section 4.3.

Bower emphasized an important kind of taxonomy in computing education: the ten task types from "declarative tasks" and "comprehension tasks" to "solve-a-problem tasks" and "self-reflect tasks" (Bower, 2008). We believe that this learner-oriented taxonomy can be a supplement to Cassel's teacher-oriented ontology.

3 DEVELOPMENT OF ONTOLOGY-BASED APPLICATIONS

We have developed a number of online applications that retrieve information for different purposes from a set of ontologies. These applications share a common graphical user interface (GUI) that facilitates navigation in the ontology and selections of the items in the ontology. This GUI is a tree-view browser.

3.1 The Tree-view Browser

A difference between our tree-view browser and many ontology tools currently available on the market is that we display not only the subclasses in the hierarchy, but also the relationships that connect to other classes in the same tree view. For instance, in Fig. 2 in addition to show class Asset's subclasses "Human" and "Countermeasure" as the "S" nodes, the relationship "assetHasVulnerability" is also displayed as an "R" node. Under class "Countermeasure", nine relationships such as "employedAt" and "isContainedIn" lead to seven targeting classes such as "AssetLifeCyclePhase" and "Product" respectively in Fig. 1. In the tree view, the "S" nodes denote the subclasses; the "R" nodes denote the range classes (the classes connected through a relationship). The check box associated with each node allows the user to select the class. Clicking on a class name, its attributes will be displayed in a panel (not shown in Figure 1).

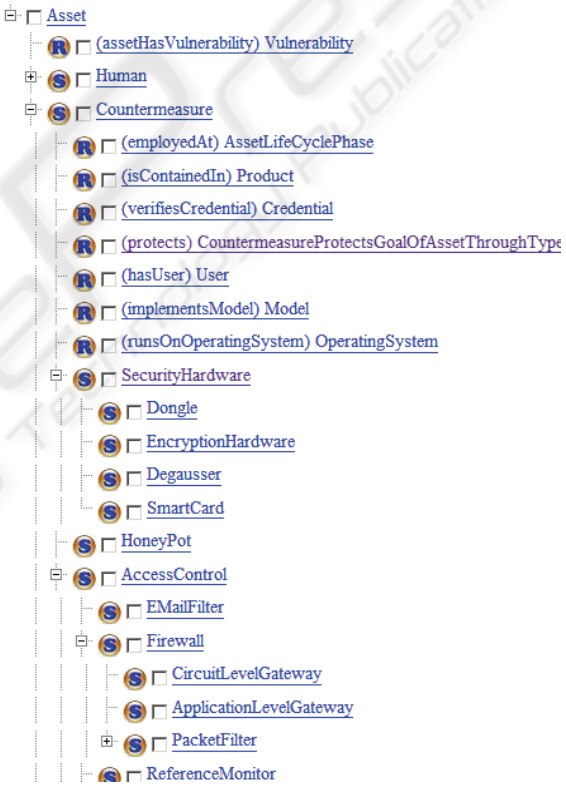


Figure 1: A tree view of a fragment of the security ontology.

Having the class hierarchy displayed, the user performs a concept-based search (Vieira, 2009). Having the relationships displayed along with the "hierarchy", the user performs a concept-based and workflow-based search at the same time. This is much more informative to the viewer who is not familiar to the given ontology, compared to Protégé that provides relationships view (property view) in a view separated from the class hierarchy.

While the ontology is presented in a tree view, the data structure may contain loops since we have included the property classes. A range class pointed by a property (relationship) is possibly a class that already appeared in the earlier part of the tree. Thus, the algorithm controlling the tree view display marks recurrence of classes in order to avoid forming any infinite loop.

The "Generate" button shown at the bottom of Figure 2 is the controller that can trigger an action according to the applications. A typical action is to generate an XML file that records the selection of classes by the user, which will be the metadata of the follow-up processing.

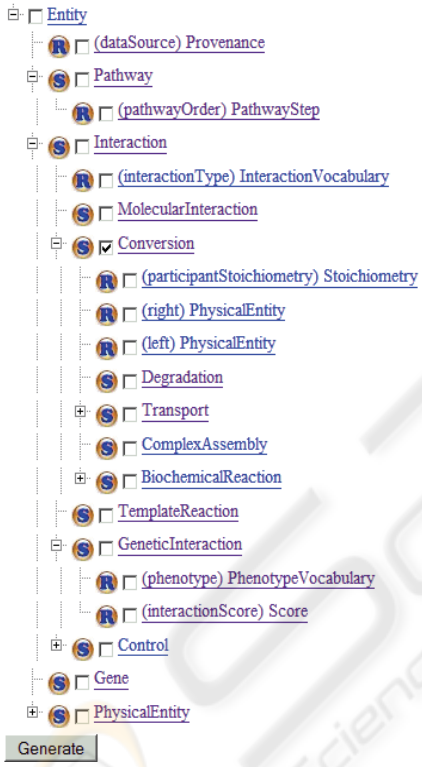


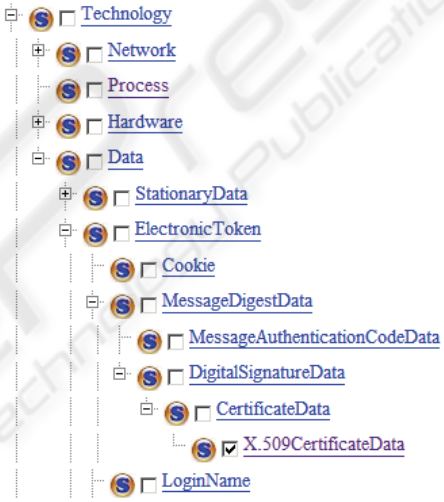
Figure 2: Instant display of a class's definition.

3.2 Biology Study Guide Producer

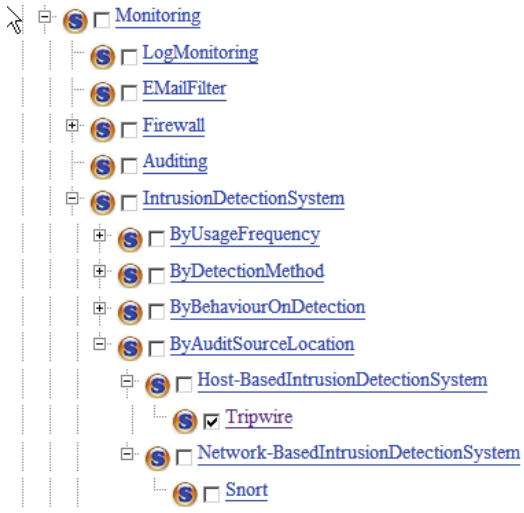
This is an application for biology professors to make customized study guides for students in learning gene-related subjects. In the bioinformatics fields, numerous high-quality ontologies have been established (EBI, 2009). For example, every class in the gene ontology (http://www.biopax.org/release/biopax-level3.owl) has a very informative comment section which is organized in multiple pieces such as "Definition", "Comment" and "Examples".

The biology professor navigates in the gene ontology as shown in Figure 2, and selects the concepts that need to be included in the study guide. In this application, we enabled the event handler of the class nodes. Upon clicking on each class node, the browser will display the definition, synonyms and examples in the comment of this class. Thus the professor can have a preview of the choice. By checking the checkbox associated to the classes, the professor can select any number of concepts. Finally, by clicking the "Generate" button, an e-book of the selected terms and related information is generated in the PDF format.

3.3 Network Security Training Materials Assemble



(a): A selected technology in the security ontology.



(b): Another selected technology in the security ontology.

Figure 3: Selected technologies for their training needs.

Computer network security has been a highly demanded area of competencies. This is a highly specialized and highly complex area. Depending on the context (the hardware, operating systems, networks, communication protocols) of the system that holds the task, the required competencies can be very diverse. In such a case, specifying the training needs of the given task can require a comprehensive and deep understanding of the knowledge and skills in the security area, which can be a significant challenge to training managers. Using the security ontology shown in Figure 1, we have produced a Web application that helps the training managers to specify the training needs.

For example, a specific task deals with "X.509CertificateData" and "tripwire". The training manager first finds these two targeted technical topics in the network security ontology by navigating in the tree view or by searching the strings. As shown in the two fragments of the tree view in Figure 3, the two topics, "X.509CertificateData" and "Tripwire", are selected.

In this application, the "Generate" button (not shown in Figure 3) will produce a list of topics consisting of the name of every ancestor node of the two chosen topics ("X.509CertificateData" and "Tripwire"), except for those nodes in the top three levels. Therefore, the training needs are made of two groups: (a) "X.509CertificateData", "DigitalSignatureData", "MessageDigestData", "CertificateData" and "ElectronicToken"; (b) "Tripwire", "Host-BasedIntrusionDetectionSystem", "ByBehaviourOnDetection", "Monitoring", and "IntrusionDetectionSystem".

3.4 METOC Training GUIs Producer

This example came from a project for the training needs of the Meteorology and Oceanography (METOC) community (Navy, 2009). The weather and ocean data are utilized by vastly diversified user applications that consume a wide range of METOC data including gridded forecast model data, climatology, weather effects data, raw satellite data, space environment and solar, remote-sensed observations, as well as imagery (METOC, 2009).

A basic and critical training need is to assure the data entry personnel are able to provide the applications with correct data. Ideally, a set of GUIs would be able to mimic the data entry scenarios of the in-field tasks of data entry. However, there are simply too many varieties of data request scenarios. Furthermore, developing highly customized METOC GUIs is very costly because it is typically difficult for the GUI programmers to understand the highly specialized METOC terminologies and distinguish synonyms and antonyms in different contexts for many tasks. We developed an ontology-based application that automates the generation of highly-tailored training GUI components. The ontology that supports this application includes a business-level ontology (BO) which is mostly consistent with the typical workflows.

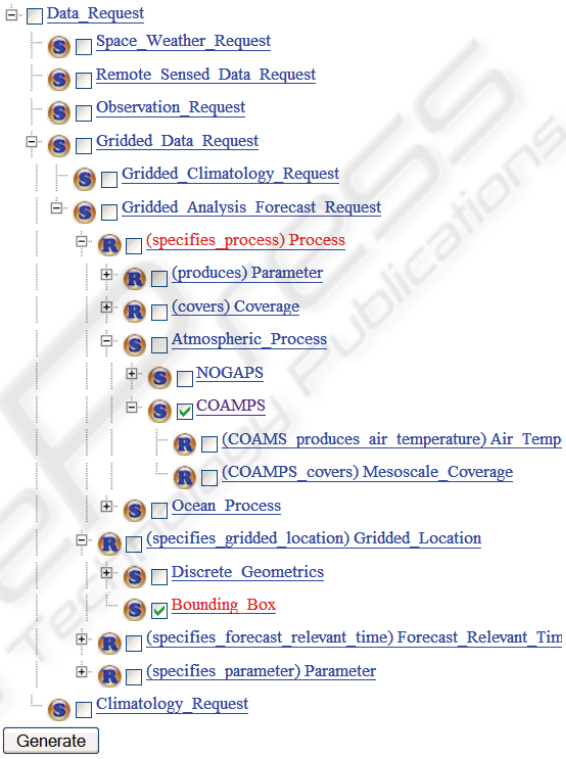


Figure 4: The tree-view of the METOC ontology.

To make the training GUI for a data entry task, the METOC engineer (the domain expert) navigates in the BO facilitated by our tree-view browser and selects the nodes of the needed data items as shown in Figure 4. For this complex application, an additional feature was added to the data selection process. Upon the expert's clicking of a class node, all its attributes are displayed in a panel on the right-hand side pane. The expert can deselect any attribute that is not needed; by default every attribute is selected. In Figure 4, two classes are selected: COAMPS and Bounding_Box. In this selection process, the domain expert can effectively specify the requirement of the training GUI since the navigation is in the language of his/her domain knowledge. As a result, the identifiers of the selected ontology classes, as well as the selected attributes of

these classes, are recorded in an XML file which comprises the metadata of the training GUI Web application.

Upon invoking the training GUI application, the attributes of the selected nodes are displayed in a tabular format. Once the user fills in and submits the form, the data is saved as "individuals" (instances of the class) in the OWL file by the Web application on the server side. The GUI for classes COAMPS and Bounding_Box are shown as Figure 5. As illustrated, not only the attributes of the chosen classes are displayed and collected, but the attributes of every superclass are also included.

4 IMPLEMENTATION

4.1 Implementation of the Tree View Browser

We have used the Jenkov JSP Tree Tag library (http://java-source.net/open-source/jsp-tag-libraries/prize-tags) to iterate over the classes and build the tree. The tag library consists of a tree model API. The tree tags display the tree model in the JSP page. Firstly, a tree instance is created.

```
ITree tree = new Tree();
```

The tree instance is to hold the information about what nodes are expanded or selected. This information is not kept in the tree nodes. Secondly, a root tree node is created and it is passed to a method called drilldown by which the child tree nodes are added recursively. Finally, the tree instance is given the root of the tree to be displayed.

```
onto1 = new OntAccess(filePath, true);
ITreeNode root = new TreeNode(c.getLocalName(),
                    c.getLocalName(), "root");
```

The user may drill down the entire tree by expanding the nodes over the Web. When he/she clicks on a particular node, the related information is displayed in a panel on the right-hand side pane. The information to be displayed can vary depending on the application. For instance, definitions and examples are displayed in the case of Section 3.2 where the nodes represent different terms and concepts in the field of genetics.

The tree-view browser is the user interface that connects to the ontology model residing on the server side. This ontology model has the following six methods which are implemented using Jena API (http://jena.sourceforge.net/ DB/index.html).

a. List all the subclasses of each object:
String[] getSubClasses (OntClass C)

b. Get Object Property of each object:
OntProperty[] getObjectProperties(OntClass C)

c. Get range classes of a property:
OntClass[] getRangeClasses(OntProperty p)

d. Return Attributes of a class:
ArrayList getAttributes(OntClass C)

e. Get all of the superclasses:
OntClass[] getSuperClasses(OntClass C)

f. Get individuals of a class:
ArrayList getIndividual(OntClass C)

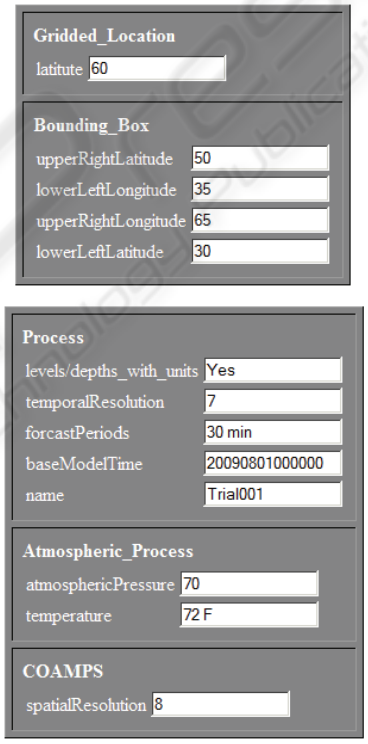


Figure 5: The GUI for METOC training.

4.2 A Note about Overlapping Local Additions to a Public Ontology

While modifying a commonly accepted public, global ontology should go through a rigorous approval procedure in order to preserve its authority, local additions to an ontology should be allowed instantly. For instance, having accessed the Computing Ontology, a computer science department wants to specify the prerequisite relationships among many topics according to the curriculum of this department. One way to add the

prerequisite relationships to the local copy of the given ontology is to create a property called "requires" in the OWL file of the ontology directly, and then add each prerequisite relationship between two topics as a sub-property of property "requires" with proper domain and range classes.

A problem will happen when the computing ontology is updated. How can we keep the local copy up-to-date without manual modifications? A solution is to start with an empty local OWL file, which then imports the remote computing ontology into this local ontology. Then the addition of those properties that represent the prerequisite relationships will be preserved in the local OWL file. When the remote ontology is updated, the next loading of the imported ontology will be the latest one.

5 CONCLUSIONS

We expect to see more applications showing the utilizations of ontologies by all kind of users and for diversified purposes. It is certain that numerous utilizations will in turn stimulate a fast growth of the ontologies themselves. Compared to any other industry, education would most benefit from the applications of ontologies because knowledge management and information retrieval are the two central activities in education.

The reported three applications are in the category of information retrieval. For these kinds of applications, our tree-view browser has demonstrated to be suitable. The tree-view browser can comfortably load the current Computing Ontology. The user will not be overwhelmed by the large number of nodes (classes, or concepts) because initially most of the nodes are folded. The user has to open the choice of nodes and then the tree view shows the result of drill down. Thus as the user navigates in the ontology, she/he is also controlling the visibility scope. Level by level, a user knows where the current focus is. So far our tree-view browser is capable of handling all the popular ontologies.

The ontology can be so large that the computer memory runs out, typically due to many direct and indirect imports of other ontologies. For that case, we have used Jena API's database persistence capability. That is, the ontology model is held in a database rather than all residing in the memory. The form of view will also be changed. We have developed a multiple-column browser similar to MSpace (http://mspace.fm/).

ACKNOWLEDGEMENTS

This work is partially supported by the United States National Science Foundation grant CCF-0939108. We sincerely thank Dr. Roy Ladner at Naval METOC Command, U.S. Navy, for his technical guidance in the METOC project.

REFERENCES

Berners-Lee, T, Hendler, J., Lassila, O., 2001. The Semantic Web, *Scientific American*, the May issue.

Bower, M., 2008. A Taxonomy of Task Types in Computing, ITiCSE'08, 281-285.

Cassel, L., et al, 2007. The Computing Ontology – Application in Education, Work group reports on ITiCSE, 171-183.

Davies, G., Cassel, L.N., Topi, H., 2006. Using a Computing Ontology for Educational Purposes, Proceedings of ITiCSE, Bologana, Italy, 334.

EBI, 2009 retrieved. European Bioinformatics Institute (EBI), Ontology Lookup Service, http://www.ebi.ac.uk/ontology-lookup/ontologyList.do

Happel, H-J, Seedort, S., 2006. Applications of Ontologies in Software Engineering, *Proceedings of the 2nd international workshop on semantic Web enabled software engineering* (SWESE), Athens, GA.

Khan, J, Hardas, M., 2007. Observing Knowledge Clustering for Educational Resources Using a Course Ontology, K-CAP'07, Whistler, B.C., Canada, 193-194.

METOC, 2009. Joint METOC Public Data Administration, http://www.cffc.navy.mil/metoc/

Navy, 2009 retrieved, Navy Weather Forcasting, http://hcs-metoc.apl.washington.edu/ forecasting_challenges.html

Ribaudo, M., Rui, 2009. M., AULAWEB, Web-Based Learning as a Comodity, CSEDU'09, 41-46.

Karlsruhe, Germany, ACM Press, article No. 10.

Sure, Y., Staab, S., Studer, R., 2002. Methodology for Development and Employment of Ontology based Knowledge Management Applications, *SIGMOD Record*, 31:4, 18-22.

Vieira, A.C. de M., Cruz, S.M.S, 2009. Semantic Annotations and Retrieval of Pharmacobotanical Data, CSEDU'09, 333-338.