AN AJAX APPLICATION TO IMPROVE ONLINE ONTOLOGY BROWSING

Ontology Explorer

Matteo Busanelli, Massimo Bondi and Nicola Gessa
ENEA Bologna, Via Martiri di Monte Sole 4, Bologna, Italy

Keywords: OWL, Ontology, AJAX, GWT, Browser, Semantic Web, Web Usability.

Abstract: OWL ontologies are one of the fundamental bricks of the Semantic Web architecture. In spite of this, ontology visualization on a web browser is nowadays not so easy; commercial browsers just show ontologies in plain text, with the OWL code. To avoid this, many ontology developers have implemented, together with their own ontologies, specific web applications to display them. But this can not be the right direction: a common Internet user does not have an easy way to browse an online ontology. Starting from a wide analysis of these applications for ontology visualization, considering their limitations and the last solutions for the development of dynamic web tools, the authors propose a GWT based architecture that, exploiting AJAX technologies, can ease the interfacing of web users with OWL ontologies.

1 INTRODUCTION

Starting from the first paper that delineated the vision of the Semantic Web (Berners-Lee et al., 2001) many efforts have been done to produce specifications, technologies, guidelines and so on to develop this vision, where data could be understood by automatic applications. The architecture of the Semantic Web nowadays is strongly based on the definition of a set of technologies, (XML, RDF, etc.). Among these, OWL, a language for ontology definition, is surely one of the fundamental.

Ontologies have been defined in different ways. One of them is "An ontology is an explicit specification of a conceptualization" (Gruber, 1995); ontologies should then represent the "knowledge" below plain data. They can perhaps be considered as the meeting point between the human vision and the automatic applications. For this reason new tools should be provided to help users that are non expert in software or informative tools, to approach the Semantic Web technologies.

Starting from these considerations, our work targets to the development of a web-based tool that could represent the right support for those that do not know (and that also do not want to know) anything about the technologies underling ontologies, but that can obtain from them serious advantages in their activities. After the analysis of the limits of the main common software for ontology management, we present an innovative web application, named Ontology Explorer, for the ontology browsing that, avoiding installation or configuration activities, aims to improve the usability and comprehensibility of ontologies that are published online. In our perspective, ontology browsing is a relevant feature that should be well supported by a software architecture that aims to improve the ontology and Semantic Web usability.

In the next section we will present the state of art about web applications for ontology browsing. Section 3 will present our design approach for the development of the application. In section 4 we will more deeply describe the implemented interface and the functionalities, detailing the innovative aspects we have introduced in respect of other solutions.

In chapter 5 we will describe our first conclusions and the future developments.

2 WEB APPLICATIONS FOR ONTOLOGY BROWSING

The first consideration starting this work was that ontologies are written in OWL file, and this format is not well supported by browsers like Internet Explorer™ or Opera™, that are not able to represent OWL files in a convenient manner, but instead in

Busanelli M., Bondi M. and Gessa N. (2009).
AN AJAX APPLICATION TO IMPROVE ONLINE ONTOLOGY BROWSING - Ontology Explorer.
In *Proceedings of the International Conference on Knowledge Management and Information Sharing*, pages 83-88
DOI: 10.5220/0002270200830088

general show the plain XML representation.

Studying the state of art, our main references have been a set of web applications for ontology browsing and Protégé, which is a widespread java-based desktop application for ontology management.

Regarding the web applications, we have selected and analysed a set of 9 products. They are:

- **Owl to Ace** (Kaljurand, 2006): transforms an OWL ontology in Attempto Controlled English (ACE). It can be considered, instead of a browser, like an application that translates an ontology from a machine-readable form in a human-readable one.
- **HarmonISA Ontology Viewer** (Hall, 2006), **Csml Ontology Viewer** (Jeong et al., 2006), **Nature Navigator** (the guide - What is Nature Navigator?, 2007), **IASCF Taxonomy Viewer** (International Accounting Standards Committee Foundation., 2005): allow the visualisation and browsing of a predefined ontology. It is based on a set of html pages linked each other.
- **RxNav** (Lussier, & Bodenreider, 2007): allows the browsing of a specific predefined ontology about pharmaceutical products. It is based on a java application that is invoked by the link on a web page.
- **Ontology Lookup Service** (Cote et al., 2006): allows the visualisation and browsing of a predefined ontology about genetics. It displays the information in a tree structure, representing a taxonomy that is the starting point to access to cards about the elements of the ontology.
- **GoFish** (Berriz et al., 2003): allows the visualisation and browsing of a set of specific predefined ontologies about genetics. It is implemented in an application that displays the information in structures like trees, cards or tables.
- **Pellet** (Sirin et al., 2003): it acts more like an ontology analyser than an ontology browser, managing any kind of ontology (RDF, OWL). It runs in batch mode, and there is no runtime interaction with the user. It allows to specify the target ontology. The analysis consists in a plain text file that reports the extracted information.
- **KSMSA Ontology Browser** (Ševčenko, 2003)**:** it is a web application for browsing the SUMO (Niles & Pease, 2001) ontology and all the connections with WorldNet, developed within the KSMSA project. In particular this tool is an online version of the more powerful editor that could be downloaded. All the ontologies are defined in the KIF (Knowledge Interchange Format) language. The browser is strictly developed for this set of ontologies.

Clearly, this list reports only the main aspects of each tool. The main considerations about these applications, starting from which we will fix a set of requirements for the Ontology Explorer, are:

- These tools do not allow the browsing of an ontology specified by the user.
- These tools are not configurable, in any perspective. This means that the user cannot customise the interface and some functions in order to make the tool more useful.
- Sometimes, the description level is too high, with too many unclear technical details, making it too cumbersome and discouraging.
- In many of them there is not the possibility to execute research operations on the ontology.

It is worth to note that the wide amount of web tools to allow Internet users to visualize online ontologies demonstrates the need for the ontology designers to show on the web in a useful manner their work. In other words, up to now, many ontology developers not only design their own ontology, but also they implement brand new "specific" tools to allow the consultation of their ontologies. In many cases this approach tries just to solve isolated issues, without considering the online visualisation of ontologies as an issue that is common for ontology designers, and without pursuing a unique solution.

After this first analysis, we concentrated on the characteristics of Protégé, that is not a web application, but that perhaps represents the main reference in the ontology definition field.

We have identified the following drawbacks in ontology browsing using Protégé (we refer to version 3.2):

- The use of labels is inadequate, and quite accessory. The browsing functionalities are based on the use of IDs, which are used to provide semantic information to the users.
- The interface for class management is separated from the interface for instance management. This presupposes in the user the knowledge of the differences between classes and instances;we want to avoid such assumption.
- Class visualisation: Protégé imposes the distinction between the logical and the property view; this is not intuitive for a common user, and again requires specific skills. Moreover, the default visualisation is the logical one, that does not display the list of the properties for a selected class, that we consider one of the fundamental information that is sought about classes.

- The research functionality acts separately on classes and instances: it is impossible to do a research on both classes and instances from the same entry point. Moreover, the research is performed only on IDs, and not on labels. Again, this is a strong limitation, and prevents the definition of multi-language ontology (since research can be performed only in one language). The research interface for instances is also laborious and not easy to use.
- Instance visualisation is not well-structured.
- The descriptions, both for classes and for instances, are displayed in tables with property/value keys, and are reported with the label rdfs:comment. Such descriptions could be instead provided in a more clear way (for example using tooltips on the labels).

Some of these limitations will be faced in the next releases of Protégé. Finally, we considered also SWOOP and Longwell, but these are not web-applications, and they are out of our context.

3 THE ONTOLOGY EXPLORER

Differently by other well-recognized widespread formats in Internet, OWL/XML files are always displayed by web browsers as plain XML files. Although we did not develop a plug-in, we wanted a way to visualize in the browsers (not in the OWL code) OWL/XML ontologies that could be found in Internet, without the need to download specific software, install it, etc. etc.

The Ontology Explorer is designed to be intuitive to use (also for the inexpert user) and many visualization and navigation configuration alternatives are available. During the design phase, we first considered the limitations listed in section 2. Then we have outlined for the tool some use cases that have been identified in different projects (Moda-ML (Gessa et al., 2005), DDTA Puglia (Regione Puglia DDTA, 2006) and Leapfrog IP (LEAPFROG-IP project)) for the development of solutions for business interoperability and we identified the generic requirements and features for the software.

3.1 Use Cases for Ontology Browsing

Without going to much in details, one of the results of the Moda-ML project has been the definition of a vocabulary of terms upon which a set of XML document schemas has been defined for data exchange in the Textile/Clothing sector (De Sabbata et al., 2005). For this vocabulary we built a semantic representation (defined with a set of OWL ontology) of both the terms and the documents defined in the vocabulary (Gessa et al., 2006).

On the other hand, the DDTA project (*Digitalizzazione dei Distretti a supporto della filiera produttiva del Tessile/Abbigliamento*) supports the definition and implementation of services to enable the constitution of enterprise networks within the textile supply chain in Puglia. This project exploits the results of Moda-ML. In these contexts domain experts are fundamental. Usually, domain experts have great knowledge about concepts that concern their expertise area, but their knowledge about ontology implementation is quite absent. Then, an example of a typical user of the Ontology Explorer (OE) could be a textile expert who consults the Moda-ML vocabulary and documents from a semantic point of view.
In this scenario of B2B interoperability:

1. It has been identified the need to search for specific concepts defined in a domain ontology in order to understand their properties and to design the proper applications and data formats to manage and exchange the related information.
2. In order to manage XML Schema documents, it has been identified the need for the designer to search for semantic concepts modelled in the XML Schema documents. In this case we need a tool to easily browse among the concepts treated in the documents, without studying the document structure. This procedure is useful to identify reusable components already defined in existing documents.

3.2 Requirements for the Tool

We have identified a set of non-functional requirements for the tool to develop. The tool

- must be able to load, read and visualize every ontology that is written in OWL.
- must hide the technical and formal aspects related to the ontology language.
- should be as much as possible interactive, allowing a fast and friendly use.
- should be configurable at different levels.
- must provide a powerful and easy to use search mechanism, regardless the way the concepts have been formalised (for example, if they are classes or instances).
- must be able to visualize classes and instances in parallel and in a homogeneous way.

The Ontology Explorer allows the loading of an OWL ontology, to browse and display every kind of

information related to the concepts defined in the ontology and to perform a search on the items of the ontology. In the following sections the capabilities of the application will be detailed deeply.

3.3 Architecture of the Application

In order to achieve the objectives fixed during the analysis phase, we adopted the AJAX technology, which allows developing highly interactive web-applications, exploiting asynchronous communication between client and server. The architecture of the tool is composed of:

- A service (server side) that implements the OWL logic and manage the loaded ontology; we called it Ontology Service.
- A set of beans-like Java objects, called OWL Wrappers, that have only to carry the information extracted from the ontology between the service and client.
- A client interface running on the web browser that is based on GWT AJAX dynamic widgets plus CSS style sheets for layout formatting.

The client will represent all the information coded in the wrappers using graphical components that we have called Semantic Widgets. These widgets are derived from the basic ones that GWT offers (like tree, textbox, etc…). They have been built to implement and exploit the logic of many OWL elements (i.e. taxonomy or relations between instances). The wrappers are used to populate the widgets with the needed content. In some cases, a different content "forces" the same widget to assume a specific behaviour: the behaviour of a widget depends on the characteristics of the content to display. The main idea behind these Semantic Widgets is to have a library of reusable graphical components that could "understand ontological structures" and to use them to better represent information and to improve the user experience.

All the Semantic Widgets were developed in a way that makes them easily graphically customizable, allowing the setting of colours, layout, images and so on simply using CSS stylesheets.

4 INTERFACE DESIGN

The interface of the Ontology Explorer (http://www.cross-lab.it/cross-lab/imple/pgcl.asp?lingua=en&p=247&node_id=6.1.1) has been structured in three different sections, in order to well separate and identify the different operations that can be performed, to reduce the confusion and the embarrassment for the user and to improve the understandability of the information.

4.1 The Loading Section

In the top of the program interface an area is dedicated to the management of the application. Here is the form to load the target ontology, to provide general information about it, to configure some aspects related to the browsing and finally to show error messages if need.

There are two ways to start the Ontology Explorer and to load a remote ontology in the tool:

- In the first way by connecting to the tool (*http://winter.bologna.enea.it/OntologyExplorer/OntologyExplorer.html*) and then specifying the URL of the OWL file in the text box.
- The second way allows to visualize in the Ontology Explorer, linked directly by another web source (like an HTML page), a specific concepts within its ontology. In other words, appending to the URL of the application, respectively the URL of the OWL file of an ontology and the name of the concept you are looking for (i.e. adding to *http://winter.bologna.enea.it/OntologyExplorer/OntologyExplorer.html* the string "*?ontologyUrl=**ontologyURL**&concept =**conceptName***"), the Ontology Explorer will be open directly on the ***conceptName.***

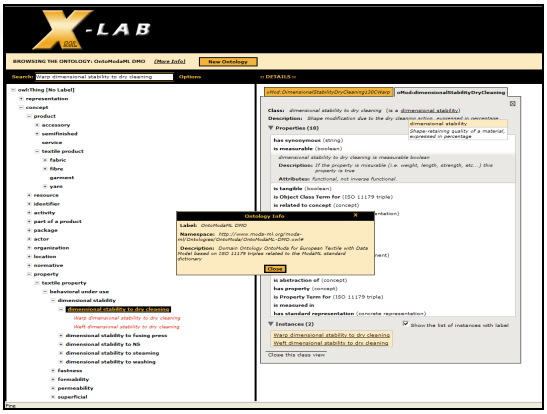

Figure 1: The interface of the Ontology Explorer.

When an ontology is already loaded (as in Fig. 1; due to space constraints, we show only a small image), here the user can find the '*More Info*' button that opens a pop-up panel showing all the general information about the loaded ontology. Also, the user can find the '*New Ontology*' button that is useful to load a new ontology.

4.2 The Navigation Section

The left part of the interface is dedicated to show a dynamic taxonomy tree used to browse and manage the taxonomies defined by the ontology. Once the ontology has been loaded, the tree is filled with all the information related to the taxonomies. The taxonomies can show both the classes and the instances or only classes. This aspect is in any case configurable (see about application configuration in the following). In this part of the interface it is also available a "search field" that allows to search in the taxonomy tree the desired concepts. The search is dynamic in the sense that while the user is typing text in the search box, if he presses the "*Enter*" key, a dynamic list of all the concepts matching the typed text is presented. In this way the user could change the search on the fly and, in many cases, he doesn't have to complete the text that he is looking for. In fact, by clicking on the list and pressing the '*Go'* button, the tree will be opened at the concept level in the taxonomy and the corresponding detail panel will be displayed in the description section. Since classes and instances are both visualised in the tree, in order to distinguish them we represent them with different formatting styles: in particular the classes are always presented in bold, while instances are presented in italic. Another important feature of the tree is the tip that, if enabled by the configuration, shows a brief textual description (if exists) of the concept every time the mouse pointer passes over it.

4.3 The Description Section

The right part of the interface is dedicated to the visualisation, in specific tab panels, of the details related to the concepts (classes or instances) selected in the taxonomy tree. In fact, by clicking an element in the tree, a tab is opened here. If the user clicks on a class the opened tab panel will show the class name (label), the description, the super class and two sub-panels: one for the list of the properties (each one containing its description and other property information like domain and range) and another for the list of the instances (each one containing its description tip and being clickable to open the relative detail panel). If the user clicks on an instance the opened panel will show, together with the name and the description, also the list of the property values for the instance. This part of the browser can collect at the same time many tabs to hold information about different selected objects. The informative elements in the panel are all clickable, in order to open directly from here other tabs and to allow a more powerful navigation between the different concepts crossing the whole ontology. Together with this interface, other kinds of information are provided using pop-up, in order to not fill the interface with too many data.

4.4 Configuration

In our perspective, an ontology can be used by different kinds of persons, with different skills. We consider very important to allow the configuration of our tool in order to match with the browsing requirements of the different users.

We then predisposed our application to provide a set of configuration parameters. For this purpose there is a panel where the user can decide if to show the classes along with their instances or not, and to show labels instead of names (OWL IDs) for a better explanation and research of the concepts modelled in the ontology. The configuration section of the Ontology Explorer consists in a pop-up that appears when the user clicks on the 'Options' button. Through this pop-up the user can configure many visualization aspects of the OE:

- To show or not instances of classes in the tree (the search will be performed also among instances accordingly to this choice).
- To show or not tooltips with the description (if any) of the concepts.
- To show concepts in the tree by their unique ontology names (ID) or by their labels.
- To set the search in "start with" (fast search) mode or in "substring" mode (slow).
- To show instances in a tab of a class by default using names or, if any, labels.

5 CONCLUSIONS AND FUTURE WORKS

Nowadays proper tools for online OWL ontology reading and browsing are lacking, or are limited.

As shown in the use cases we identified in section 3.1, Semantic Web technologies can solve a set of problems related to business vocabulary management. We developed an AJAX-based web application named Ontology Explorer that overcomes some of the limitation of existing tools and allows an intuitive ontology browsing for every kind of Internet user. In particular, the OE:

1. is able to load every OWL/XML ontology on the Internet; in this sense it is not built to manage a specific ontology. We have tested it

with many online ontologies, and with the Moda-ML ontology, that includes 373 classes, 44 relationships and 1098 instances.

2. matches with the skills of a person that is not an ontology expert. To this aims, the concepts of *classes, instances, datatype* or *object property* are completely transparent to the users. A central issue has been also the design of friendly forms for information visualisation.

3. provides a powerful research interface.

4. exploits the use of labels, that allows the browsing of multi-language ontologies, a more strong comprehension of the terms and a stronger search and browsing. The concept IDs in an ontology should be used as are primary keys in a database, just to identify a class/instance/-property, but IDs should not be used to provide semantic information (as usually happens).

It is worth to note that the Ontology Explorer distinguishes itself from other tools like Protégé because it should be considered not like an editor for Ontology Expert (like Protégé), but like a powerful browser for Domain Expert. Considering our use cases, exploiting the semantic representation of the Moda-ML business vocabulary, the tool has simplified the interfacing with the terms and the XML Schema documents, allowing a more easy search and analysis of them. The future steps to improve the Ontology Explorer will concern mainly:

- To enhance the configurability of the tool providing a new set of customizations about the domain of user's interest. This will be achieved by defining some *Semantic Profiles* that could be *adopted* by the user (externally to the ontology) to browse a specific ontology.
- To add a new *Advanced Semantic Search* that could use logic inference to deduce non explicit information and that allows enlarging the resulting set of the matching concepts to all the semantically related ones.
- To improve the graphical aspect and the usability and to solve some compatibility issues with some browsers (for instance Opera™ and Firefox™ don't render some widgets so well).

REFERENCES

Berners-Lee, T., Hendler, J., Lassila, O.. The semantic web. Scientific America, May 17, 2001.

Gruber, T. R.. Toward principles for the design of ontologies used for knowledge sharing. In International Journal of Human Computer Studies 43 (5/6), pag. 907–928, 1995.

Kaljurand, K., Fuchs, N.E.. Bidirectional mapping between OWL DL and Attempto Controlled English. Budva, Montenegro, 2006.

Hall, M.. HarmonISA Land-Use Viewer System - System Documentation and Handbook. HarmonISA Publication, July, 2006.

Jeong, E., Nagasaki, M., Saito, A., Miyano, S., Ontology-Based Representation for Cell System Markup Language. CSML Publications, 2006.

Lussier, Y.A.., Bodenreider, O.. Semantic Web: Rivolutionizing knowledge discovery in the life sciences. Springer, 2007.

About the guide - What is Nature Navigator?. The Natural History Museum, 2007.

Cote, R.G., Jones, P., Apweiler, R., Hermjakob, H.. The ontology lookup service, a lightweight cross-platform tool for controlled vocabulary queries. BMC Bioinformatics, 28 February, 2006.

International Accounting Standards Committee Foundation. About the IFRS-GP Taxonomy Viewer. International Financial Reporting Standards, General Purpose Financial Reporting for Profit-Oriented Entities, Incorporating Additional Requirements for Banks and Similar Financial Institutions (IFRSGP), May 15, 2005.

Berriz, G.F., White, J.V., King, O.D., Roth, F.P.. GoFish finds genes with combinations of Gene Ontology attributes. Bioinformatics, 2003.

Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.. Pellet: A Pratical OWL-DL Reasoner. Journal of Web Semantics, 2003.

De Sabbata, P., Gessa, N., Cucchiara, G. , Imolesi, T., Vitali, F.. Supporting eBusiness with a dictionary designed in a vertical standardisation perspective. IEEE-CEC Conference, Munich, Germany, July 19-22 2005, pp. 160-167, ISBN 0-7695-2277-7.

Gessa, N., Cucchiara, G., De Sabbata, P., Brutti, A.. A bottom-up approach to build a B2B sectorial standard: the case of Moda-ML/TexSpin. In Proceedings of INTEROP ESA 05-NOREST Workshop, Geneve February 22nd 2005, publication by Herm`es Science Publications.

Regione Puglia. Assessorato allo Sviluppo Economico, Progetto "Distretto Digitale a supporto della filiera produttiva del Tessile Abbigliamento" (DDTA), 2006.

Gessa, N., Busanelli, M., De Sabbata, P., Vitali, F.. Extracting a semantic view from an ebusiness vocabulary. IEEE International Conference on E-Commerce Technology, San Francisco, California, June 26-29, 2006, pp 398-401, ISBN 0-7695-2511-3.

Ševčenko, M.. Online Presentation of an Upper Ontology. In Proceedings of Znalosti 2003, Ostrava, Czech Rep., February 19-21, 2003, ISBN 80-248-0229-5.

Niles, I. Pease, A.. Towards a Standard Upper Ontology. 2nd International Conference on Formal Ontology in Information Systems (FOIS), Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.

LEAPFROG-IP project: http://www.leapfrog-eu.org/ LeapfrogIP/main.asp