Ontobee: A Linked Data Server that publishes RDF and HTML data simultaneously

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Zuoshuang Xiang[a], Chris Mungall[b], Alan Ruttenberg[c], Yongqun He[a,*]
[a]*University of Michigan Medical School, Ann Arbor, MI 48105, USA.* [b]*Lawrence Berkeley National Laboratory, Berkeley CA, USA, 48109,* [c]*University at Buffalo, Buffalo, NY, USA*

Abstract. The Semantic Web allows machines to understand the meaning of information on the World Wide Web. The Linking Open Data (LOD) community aims to publish various open datasets as RDF on the Web. To support Semantic Web and LOD, one basic requirement is to identify individual ontology terms as HTTP URIs and deference the URIs as RDF files through the Web. However, RDF files are not as good as HTML for web visualization. We propose a novel "RDF/HTML 2in1" model that aims to report the RDF and HTML results in one integrated system. Based on this design, we developed Ontobee (http://www.ontobee.org/), a web server aimed to dereference ontology term URIs with RDF file source code output and HTML visualization on the Web and to support ontology term browsing and querying. Using SPARQL query of RDF triple stores, Ontobee first provides a RDF/XML source code for a particular HTTP URI referring an ontology term. The RDF source code provides a link using XSLT technology to a HTML code that generates HTML visualization. This design will allow a web browser user to read the HTML display and simultaneously allow a web application to access the RDF document. The HTML display supports dereferencing of the ontology term information in user-friendly HTML format. The RDF output supports remote query of the ontology term and the Semantic Web. The contents of the HTML and RDF output files can be different. Ontobee provides a user-friendly web interface for displaying and querying the details and its hierarchy of a specific ontology term. Ontobee currently support 101 ontologies with over 1,300,000 ontology terms. It has become the default linked data server for publishing and browsing biomedical ontologies in the OBO foundry library. In summary, Ontobee provides an efficient and publicly available method to promote ontology term URI dereferencing, web visualization and query, and further facilitate the Semantic Web and LOD.

Keywords: Semantic Web, LOD, RDF, RDF/HTML 2in1, Ontobee, ontology visualization

1. Introduction

Biomedical ontologies are consensus-based controlled biomedical vocabularies of terms and relations with associated definitions, which are logically formulated to promote automated reasoning. Biomedical ontologies play a critical role in the process of achieving the Semantic Web and the Linking Open Data (LOD). The Semantic Web is a web of data that allows machines to understand the meaning – or "semantics" – of information on the World Wide Web (WWW). To ensure that computers can understand the semantics of terms, machine-readable ontologies that refer terms with ontology uniform resource identifier (URIs) are required. However, how to present the meaning of ontology terms by its URI is still a challenge. Most ontology URIs do not point to real web pages. While some URIs point to specific pages, the pages shown are often in pure HTML format or in OWL (RDF/XML) format which contains

*Corresponding author. E-mail: yongqunh@umich.edu.

the whole ontology instead of individual terms. In both cases these pages do not efficiently support the Semantic Web.

The objective of the Linking Open Data (LOD) community is to extend the Web with a data commons by publishing various open datasets as RDF links on the Web. These RDF links between data items can come from different data sources and be accessed anywhere through the web. All of the sources on these LOD diagrams are open data. Tim Berners-Lee, the inventor of the World Wide Web, outlined four Linked Data Principles: (i) Use URIs to identify things; (ii) Use URIs of HyperText Transfer Protocol (HTTP) so that these things can be referred to and looked up ("dereferenced") by people and user agents; (iii) Provide useful information about the thing when its URI is dereferenced, using standard formats such as the Resource Description Framework (RDF); and (iv) Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web [1]. The basic element of LOD data exchange is the definitions of individual terms and logical relations among these terms. This basic element can be resolved using ontologies.

To support semantic web and LOD, it is required to dereference individual ontology term URIs to real RDF documents through the Web. However, since RDF documents are plain text and not comfortable to read, it is preferred to browse and look up the detailed information about individual ontology terms using HTML format on the Web. It would also be ideal to develop a user-friendly query system for searching ontology terms. It is a challenge to integrate all these features into one system.

Many ontology visualization and searching programs exist. One commonly used ontology repository and ontology browser system is the National Center for Biomedical Ontology (NCBO) BioPortal (http://bioportal.bioontology.org/). NCBO BioPortal supports user query, ontology hierarchical tree display, and ontology term content visualization. The Ontology Lookup Service (OLS; http://www.ebi.ac.uk/ontology-lookup/) [2] also provides lookup service and browsing capability for the OBO-Format ontologies in the Open Biomedical Ontologies (OBO) Library [3]. However, OLS has limited capabilities for handing the Web Ontology language (OWL). Many LOD browsers are available, for example, Ontology-browser (http://code.google.com/p/ontology-browser/), and more listed on the following web page: http://en.wikipedia.org/wiki/Linked_Data#Browsers.

However, these ontology browsers do not provide the output of ontology annotation in the RDF/XML format, an important feature for various web applications and LOD development. Instead, NCBO provides the NCBO Resource Index [4]. The Resource Index application programming interface (API) is available via a Representational State Transfer (REST) web services interface. Based on the Java programming, this web service allows users to extract information for individual ontology terms. The web service is separated from the HTML Bioportal browser output. Special effort is needed to implement the information extraction task.

A Linked Data Server is a web server that allows the exposure of linked data following the four Linked Data Principles described above [5]. The nature of a Linked Data Server requires that such a server processes RDF efficient. Therefore, a RDF triple store is the preferred choice for implementing a Linked Data Server. However, although a RDF triple store is ideal to process RDF files, it does not naturally support HTML format for web visualization.

In this report, we first present our generation of a “RDF/HTML 2in1” model as a novel design pattern. This design pattern aims to represent both RDF and HTML in one seamless system to support the Semantic Web and LOD. Based on this design pattern, we have developed Ontobee (http://www.ontobee.org), a web-based Linked Data Server that provides RDF output to each ontology term IRI and simultaneously, it generates user-friendly browsing and querying of ontology term contents through the HTML format.

2. Methods

2.1. New design pattern development

The “RDF/HTML 2in1” design pattern is generated during our process of developing Ontobee with an initial focus of using SPARQL queries to support browsing of Vaccine Ontology [6] and then other OBO foundry library ontologies. It combines an RDF output for web applications with a HTML output for web browsing and searching. Such a design pattern is a summarized pattern out of our software development process and frequent discussions.

2.2. Ontobee server and programming tools

The Ontobee application server is implemented using one HP server running Red Hat Linux operating system (Red Hat Enterprise Linux 5 server). The open source software program, Apache HTTP Server, is installed as the HTTP application server. PHP is used as programming languages in the web application server.

2.3. Ontology source used in Ontobee

The primary ontology source comes from the He-group RDF store (http://sparql.hegroup.org/sparql), a knowledgebase containing all the ontologies in OBO foundry [7]. This RDF store use Virtuoso Open-Source Edition as the backend. A pipeline consisting of a list of PHP/Java scripts was developed to retrieve OBO ontologies, merge them into single OWL files, and then load the OWL files into the RDF store. Ontobee also supports ontologies from other data sources. For example, Ontobee also maintains a Virtuoso RDF store that contains several ontologies such as the Interaction Network Ontology (INO) (http://sourceforge.net/projects/ino/).

2.4. Query of RDF triple store and process of queried data for individual ontology terms

A software program based on PHP and SPARQL was developed to visualize the hierarchical tree and individual terms from a specific ontology. The ontology tree is assembled from the results of a set of SPARQL queries against the RDF stores. The 'transitive' option in the Virtuoso SPARQL engine was used to minimize the number of SPARQL queries. For individual terms, a HTML page is assembled from the results of another set of SPARQL queries against the RDF stores. The 'transitive' and the 'CBD' (i.e., Concise Bounded Description) option in the Virtuoso SPARQL engine were used to minimize the number of SPARQL queries. In both cases, PHP is the primary programming language, and it interacts with the RDF stores by sending SPARQL queries to the stores. The stores then return the results back to PHP scripts in JSON (JavaScript Object Notation; http://www.json.org/) format. The PHP scripts decode information from the JSON formatted results and format it into an RDF output file and a linked user-friendly HTML page.

2.5. Output of RDF and HTML simultaneously associated with individual ontology terms

Upon a query of an ontology term URI, Ontobee outputs the contents of each ontology term with both RDF and HTML formats. This OWL (RDF/XML) format is the default page format for the term URI. The HTML file is embedded in a XSLT (Extensible Stylesheet Language Transformations) [8] template which is linked to the OWL file. When the term URI was visited by using a web browser, both the OWL file and the HTML content will be retrieved but only the HTML content will be shown in the browser for easy reading. A user can easily access the OWL (RDF/XML) content by check the source code of the term page. Alternatively, the OWL output file can be retrieved by a web application program, or a Semantic Web system. This approach does not need a human visit to the HTML web page.

2.6. Development of Ontobee ontology term searching program

The jQuery JavaScript Library (http://jquery.com/) is used for the development of Ontobee ontology term searching program. This library provides autocompletion feature to the searching function. This program is available in the Ontobee cover page to search terms from all ontologies. It is also available in the home page of individual ontologies in Ontobee for searching terms from single ontologies.

2.7. Software download and license

The source code of Ontobee can be freely downloaded from sourceforge.com: Project site: http://sourceforge.net/projects/ontobee/. The license for this software reuse is the Apache License 2.0.

3. Results

3.1. The "RDF/HTML 2in1" model for publication of linked ontology term data

We have generated a "RDF/HTML 2in1" model to describe the ontology term information publication (Fig. 1). The central idea of this design pattern is to deference ontology term URIs with two outputs:

- RDF output for web applications;
- HTML output for web visualization and search

A RDF triple store system (*e.g.*, Virtuoso http://virtuoso.openlinksw.com/) is a preferred data management system to store and process linked ontology data. To generate both outputs, we can use two methods:

- To first generate RDF output and then embed an HTML link in the RDF file. This is the approach currently used in Ontobee. A method for this strategy is to use the XSLT (Extensible Stylesheet Language Transformations), a declarative, XML-based language used for the transformation of XML documents. XSLT is able to generate a new document based on the content of an existing one. The new document may be serialized (output) by a server in another format such as HTML (Ref. http://en.wikipedia.org/wiki/XSLT). XSLT is often used to convert XML data into web pages or PDF documents.

- To first output an HTML file and then provide link to RDF output inside the HTML file (Fig. 1). In this case, XSLT is not needed.

RDF is a preferred method for dereferencing URI and providing useful information in the LOD community [5]. Therefore, the first method of outputting RDF results and then including embed HTML link in the RDF file is a preferred method for the LOD.

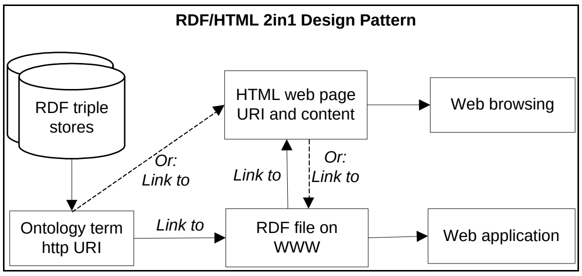

Fig. 1. The design pattern of the "RDF/HTML 2in1" model

3.2. *Ontobee software design and architecture*

Ontobee has been developed based on the "RDF/HTML 2in1" design pattern. The Ontobee software integrates two basic features: one is RDF content sharing, the other is ontology content visualization (Fig. 2). Basically, a user can use a web browser to query Ontobee web page. Alternatively, a web application can be developed to access the RDF output file directly without using a web browser. Ontobee will then issue SPARQL queries to a RDF triple story. Initially, Ontobee used the Neurocommons RDF triple store that is not continuously maintained and updated [7]. In addition, this Neurocommon triple store uses unmerged ontologies. This leads to the ignorance of all imports when the base ontology is queried from the triple store. Therefore, the usage of the unmerged version makes the data incomplete (*i.e.*, the imported data missing from the base ontology) and slower query performance. Currently Ontobee uses the Hegroup Virtuoso triple store that takes a merged version of any particular ontology, *i.e.*, merging all imports into the base ontology. The Hegroup triple store also used updated data sources and loading system. The results of the SPARQL queries are returned in JSON format and processed by Ontobee PHP scripts to form a user-friendly HTML page. Simultaneously, RAW OWL output was also returned and used to generate machine-readable RDF output file.

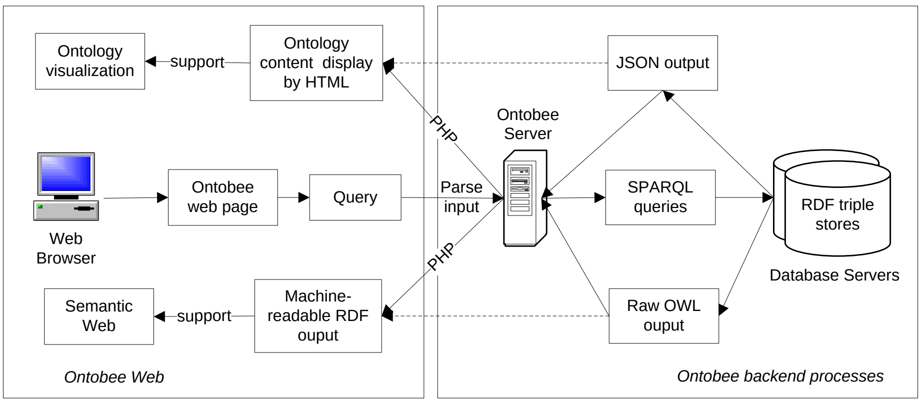

Fig. 2. Ontobee system architecture design.

3.3. *Ontobee statistics*

Currently Ontobee provides access to 101 ontologies with over 1,300,000 ontology terms (Table 1). The majority of these ontologies come from the OBO foundry Library (Reference: http://wwww.obofoundry.org) [3]. Ontobee also includes many ontologies (*e.g.*, Cell Line Ontology or

CLO) which are not yet included in the OBO Foundry library. These ontologies cover different domains, such as anatomy, health, and experiments.

Table 1. Summary of selected ontologies covered in Ontobee.

Ontology Names	Number of terms
OBO foundry ontologies and candidate ontologies	
GO (Gene Ontology)	36,212
PR (Protein Ontology)	28,711
PATO (Phenotype Ontology)	2,302
IAO (Information Artifact Ontology)	244
IDO (Infectious Disease Ontology)	549
OBI (Ontology for Biomedical Investigation)	3,616
Cell Type Ontology	4,133
VO (Vaccine Ontology)	5,488
OAE (Ontology of Adverse Events)	1,154
NCBITaxon (NCBI organismal classification)	847,760
Other biomedical ontologies	
INO (Interaction Network Ontology)	1,018
Cell Line Ontology (CLO)	38,184

3.4. Ontobee as a linked RDF data server

When a web browser user or a web application searches Ontobee for an ontology term URI, for example, the term ‘vaccination’ from the Vaccine Ontology (VO), a PURL program in the obolibrary.org will direct the URI to an Ontobee URI, *e.g.*, http://www.ontobee.org/browser/rdf.php?o=VO&iri=http://purl.obolibrary.org/obo/VO_0000002. A Uniform Resource Identifier (URI) is defined in [RFC3986] as a sequence of characters chosen from a limited subset of the repertoire of US-ASCII [ASCII] characters. The Internationalized Resource Identifier (IRI) extends the syntax of URIs to a much wider repertoire of characters by using the Universal Character Set (Unicode/ISO 10646) (http://www.ietf.org/rfc/rfc3987.txt). In Ontobee, IRI is used instead of URIs to identify resources. Ontobee uses SPARQL to query the RDF triple store and develops a PHP script to trace and parse the RDF raw data and display the results in a RDF/XML file.

If the source page of an Ontobee web page that dereferences this IRI is displayed, the RDF/XML file can be shown (Fig. 3). This file is a valid OWL-DL that supports inference and reasoning. The second line of the RDF/XML file contains a tag as follows:

```
<?xml-stylesheet type="text/xsl" href="/browser/xslt.php?o=VO&amp;iri=http://purl.obolibrary.org/obo/VO_0000001"?>
```

This line specifies an XLST stylesheet in which the HTML code is embedded. In the example of our Ontobee implementation, the URL of the XML stylesheet is constructed to specify the label abbreviation of a particular ontology (*e.g.,* VO) and an IRI of a specific term (*e.g.,* the full URI for the VO term VO_0000002 for the label ‘vaccination’) in this ontology. The HTML code is embedded in this XLST stylesheet.

The RDF/XML output file used in Ontobee is an OWL format derived from SPARQL queries of data from a RDF triple store. An RDF file can be parsed down to a list of triples. A triple consists of a subject, a predicate, and an object. The subject identifies what object the triple is describing. The predicate defines the piece of data in the object to be given a value. The object is the actual value. OWL extends RDF with additional predicates to characterize properties and classes. SPARQL is a query language for querying RDF triple stores. Ontobee mainly using the SPARQL ‘describe’ function. To minimize the file size, the raw output from SPARQL queries of RDF triples associated with a specific ontology term was reformatted by Ontobee. An example of the RDF/XML output syntax is as follows:

```
<rdf:Description rdf:about="&obo;VO_0000001"><rdfs:label>vaccine</rdfs:label></rdf:Description>
```

This syntax represents that the VO term IRI VO_0000001 has a label of ‘vaccine’. The rdf:nodeID seen in Fig. 3 is the information from the RDF triple store that is not typically useful by users.

Fig. 3. Demonstration of an Ontobee RDF output file. This is also part of the source page view of the ontology term http://purl.obolibrary.org/obo/VO_0000002 (label: 'vaccination' from VO).

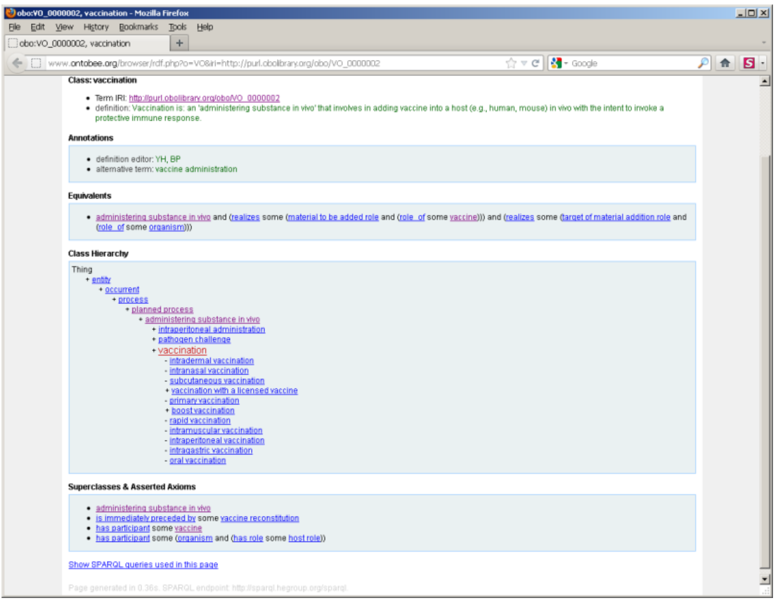

Fig. 4. Demonstration of an Ontobee HTML web visualization of the VO term 'vaccination' in Firefox web browser.

3.5. Ontobee for visualizing and querying ontology term information based on HTML

The HTML web page contains a user-friendly display of the term information, including all information about this term (Fig. 4):

- Basic Term IRI and term definition.
- Term annotations, including definition editor, editor notes, and other annotations such as seeAlso.
- Term equivalents. A term equivalent represents a logic definition of the term with sufficient and necessary conditions.
- Class hierarchy. This hierarchy includes the term's ancestor terms at different levels and its immediate child terms as well. A sign "+" next to a term means that this term has child term(s). A sign "-" indicates that a term has no child term.
- Superclasses and asserted axioms. In ontology, axioms are used to associate class and proper-

ty IDs with either partial or complete specifications of their characteristics, and to give other logical information about classes and properties. A class axiom contains a collection of descriptions, which can be more-general classes, restrictions, sets of individuals, and boolean combinations of descriptions.

- Usages. This section includes the list of all usages associated with this term.
- Disjoints. Two ontology terms are disjointed if their contents do not overlap.

Each ontology term shown in this page is clickable (Fig. 4). Once a term is clicked, the detailed information about the ontology term will be displayed in another corresponding page.

All the results are obtained from internal SPARQL queries of a RDF triple store. In the bottom of the page, a grey line of words indicates how many SPARQL queries have been conducted to generate this web page.

3.6. Ontology term searching in Ontobee

Ontobee provides a keyword searching program. This search feature on the Ontobee cover page can be used to query all ontologies or any selected ontology listed in Ontobee (Fig. 5). Ontobee uses jQuery, a fast and concise JavaScript Library (http://jquery.com/). The jQuery library simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. When a user types some keyword(s), Ontobee search program will list all hit terms in an alphabetic order in a drop-down menu (Figure 5A). This is full-text search, meaning that it only finds completed words. The matched words do not have to start from the beginning of an ontology term. Incomplete words will not be identified. After the user selects a term from the menu, the detail about this term will show up in another page (Fig. 5B). Alternatively, after a user types a query keyword(s), the user can click on “Search terms” next to the keyword(s). This operation will result in the listing of all queried results in alphabetic order on another web page (Fig. 5C). On the home page of each individual ontology, the same searching function is available, but only associated with the single ontology

3.7. Ontology usages and performance evaluation

Historically, Ontobee was first developed to visualize the Vaccine Ontology (VO) [6]. Its development has obtained much attention and feedback from the ontology community, especially the OBO Foundry ontology development community. Ontobee was then used as the default Linked Data Server for the Ontology for Biomedical Investigation (OBI) [9]. Ontobee has now been used as the default Linked Data Server for all ontologies listed in the OBO Foundry Library. The internal redirection through the PURL system is performed by the OBO Foundry library administrators. In addition, Ontobee has been used for several ontologies that are not yet listed in OBO Foundry library.

A performance evaluation was conducted on March 23, 2012. Briefly, we randomly chose a list of ontology term URIs and queried the URIs in Ontobee. Part of the results was reported in Table 2. The query speed was fast, typically within 0.1 – 2 seconds for each ontology term query in Ontobee. The speed depends on many factors, including the number of SPARQL queries, the number of usages for each ontology term, the number of associated axioms, and whether or not the pages were loaded recently.

Currently, Ontobee has also been used in various web applications. The Ontobee RDF triple store has been used in OntoFox, a web server that supports ontology reuse by allowing users to input terms, fetch selected properties, annotations, and certain classes of related terms from source ontologies, and save the results using the RDF/XML serialization of the OWL [10]. The Ontobee RDF triple store has also been used for development of Ontodog (http://ontodog.hegroup.org), a web application that generates ontology views. The RDF output of Ontobee has also been used by web applications. For example, the Minimum Information Required in the Annotation of Models Registry (http://www.ebi.ac.uk/miriam) automatically detects and monitors individual Ontobee RDF outputs for all ontology terms of Ontobee-listed ontologies [11].

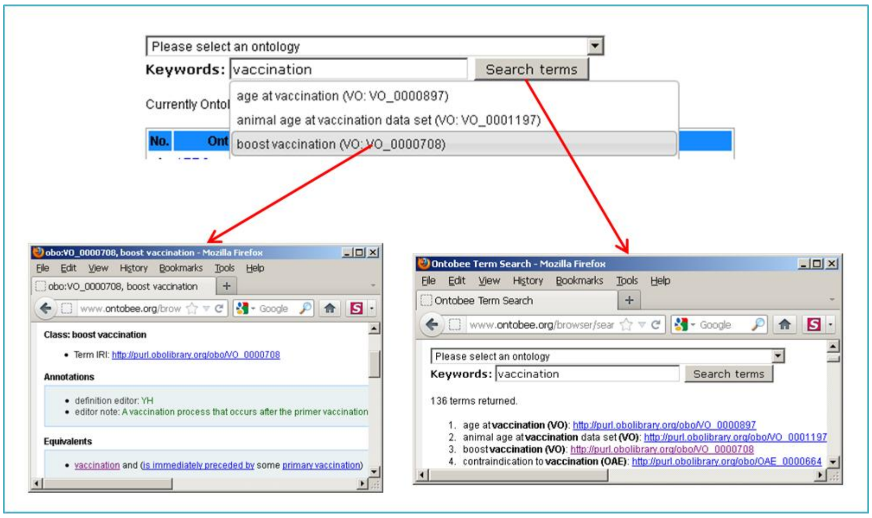

Fig. 5. Demonstration of a HTML web visualization of the VO term 'vaccination' in a Firefox web browser.

Table 2. Speed of randomly selected Ontobee queries (Performed on March 22, 2012)

Example Ontology Terms	Speed
obo:VO_0000002, vaccination	1.46 sec for 11 SPARQL queries, 362 usages
obo:UBERON_0000972, antenna	0.20 sec for 10 SPARQL queries, 1 usage
obo:OBI_0000070, assay	0.65 sec for 11 SPARQL queries, 76 usages
obo: NCBITaxon_9606, homo sapiens	0.21 sec for 8 SPARQL queries, 0 usage
obo:CL_0000000, cell	0.12 sec for 7 SPARQL queries, 0 usage
obo:IDO_0000596, infectious agent	0.29 sec for 11 SPARQL queries, 21 usages

4. Discussion

The problem of identity and reference on the web has been discussed for long in and outside WWW circles. Our novel "RDF/HTML 2in1" model provides a way to run both RDF output for web services/applications and HTML output for web visualization and query of individual ontology terms. Based on this novel design pattern, Ontobee is developed with an attempt to unify biomedical ontology visualization and ontology term transfer based on RDF/XML format. With the Ontobee service, each ontology term URI has its own page and contains a term-specific RDF file. This RDF output format is especially useful for those web applications that use RDF input format. Meanwhile, based on the XSLT technology, a separate HTML web page is demonstrated for web browser users who can visualize the details about this ontology term in a user-friendly format. To our best knowledge, Ontobee is the first program that implements such a strategy of outputting both RDF and HTML results simultaneously. Ontobee provides a useful tool to support the Semantic Web and the LOD as well as user-friendly browsing and searching.

Our Ontobee system differs from the NCBO Bioportal and NCBO Resource Index in terms of ontology term browsing and URI dereferencing [4]. The NCBO Bioportal and Resource Index use Java technology and a REST web services interface [12]. In contrast, our Ontobee system uses PHP programming and SPARQL for querying RDF data from a RDF triple store. RDF can be used to create Web services, both wired and wireless, for metadata, or data that is maintained by an application [13]. The output of RDF provides a way for web applications to directly use the RDF output for further applications. Another difference is that the NCBO approach separates the ontology web browser and RDF output using NCBO Bioportal and Resource Index, respectively. In contrast, Ontobee combines these two features together in a lightweight pattern. The Ontobee method is much easier to implement. The speed of Ontobee query is fast.

Ontobee is highly scalable. Although its current usage is primarily on OBO ontologies, Ontobee can also be applied to non-OBO ontologies. The Ontobee software can be freely downloaded and used. Its maintenance over time is guaranteed with the community support.

More Ontobee features are expected to be developed. For example, the RDF output format can be improved with suggestions from Ontobee users and developers. Currently, the Ontobee HTML is not fully styled. We are aiming to make it fully styleable with the Cascading Style Sheets (CSS) so that ontologies could supply customized CSS. More applications of Ontobee are also being discussed and designed.

5. Acknowledgments

Development of Ontobee is supported by NIH grant 1R01AI081062.

References

1. Berners-Lee T (2009) Design Issues: Linked Data: http://www.w3.org/DesignIssues/LinkedData.
2. Cote RG, Jones P, Martens L, Apweiler R, Hermjakob H (2008) The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. Nucleic Acids Res 36: W372-376.
3. Smith B, Ashburner M, Rosse C, Bard J, Bug W, et al. (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25: 1251-1255.
4. Jonquet C, Lependu P, Falconer S, Coulet A, Noy NF, et al. (2011) NCBO Resource Index: Ontology-Based Search and Mining of Biomedical Resources. Web Semant 9: 316-324.
5. Bizer C, Heath T, Berners-Lee T (2009) Linked Data - The Story So Far. International Journal of Semantic Web and Information Systems 5: 22 pages.
6. He Y, Cowell L, Diehl AD, Mobley HL, Peters B, et al. VO: Vaccine Ontology; 2009 July 24-26, 2009; Buffalo, NY, USA.
7. Ruttenberg A, Rees JA, Samwald M, Marshall MS (2009) Life sciences on the Semantic Web: the Neurocommons and beyond. Brief Bioinform 10: 193-204.
8. Clark J (1999) XSL Transformations (XSLT) Version 1.0. W3C Proposed Recommendation 16 November 1999. Available at http://www.w3.org/TR/xslt.
9. Brinkman RR, Courtot M, Derom D, Fostel J, He Y, et al. (2010) Modeling biomedical experimental processes with OBI. Journal of Biomedical Semantics: Accepted.
10. Xiang Z, Courtot M, Brinkman RR, Ruttenberg A, He Y (2010) OntoFox: web-based support for ontology reuse. BMC Res Notes 3: 175.
11. Juty N, Le Novere N, Laibe C (2012) Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. Nucleic Acids Res 40: D580-586.
12. Fielding RT (2000) Architectural Styles and the Design of Network-based Software Architectures. Irviine, CA, USA: University of California.
13. Hjelm J (2001) Creating the Semantic Web with RDF: Professional Developer's Guide. New York, NY, USA: John Wiley & Sons, Inc.