The Live OWL Documentation Environment: a tool for the automatic generation of ontology documentation

Silvio Peroni[1], David Shotton[2], and Fabio Vitali[1]

[1] Department of Computer Science, University of Bologna
[2] Department of Zoology, University of Oxford
essepuntato@cs.unibo.it, david.shotton@zoo.ox.ac.uk, fabio@cs.unibo.it

Abstract. In this paper we introduce *LODE*, the *Live OWL Documentation Environment*, an online service that automatically generates a human-readable description of any OWL ontology (or, more generally, an RDF vocabulary), taking into account both ontological axioms and annotations, and ordering these with the appearance and functionality of a W3C Recommendations document. This documentation is presented to the user as an HTML page with embedded links for ease of browsing and navigation. We have tested LODE's completeness and usability by recording the success of test users in completing tasks of ontology comprehension, and here present the results of that study.

Keywords: LODE, OWL ontologies, Web tools, ontology documentation, user testing

1 Introduction

Any strategy that guarantees the broad adoption of Semantic Web technologies must address the need for improved human interaction with semantic models and data. While research has been undertaken on models, theoretical approaches and the development of tools to infer new information from data and ontologies, the Semantic Web will never be really integrated with the everyday Web until semantic information is easily accessible to ordinary users through Web browsers, not limited to Semantic Web practitioners employing specialist tools. This point is even more crucial for Semantic Publishing, since its end-users are by definition publishers, researchers, librarians and general readers, rather than experts in semantic technologies. Thus the Semantic Web / Semantic Publishing communities need to develop user-friendly Web interfaces that mediate between semantic models and end-users.

Of course, work has already been done in this direction. For instance, ontology development editors have been created (e.g. Protégé[3] [12] and the NeOn Toolkit [19]), Web search engines to look for semantic resources launched (e.g.

[3] Protégé: http://protege.stanford.edu.

Sindice[4] [15] and Watson[5] [5]), and semantic desktop applications released (e.g. SemNotes[6] [7]). However, what is still missing, and what the Semantic Publishing community urgently needs, are tools that assist people who are not expert in semantic technologies in dealing with and publishing semantic data, and in particular in understanding the ontologies that make this possible.

Human interactions with ontologies usually involves the following steps:

1. Once ontologies suitable for the particular domain of interest have been identified, people need to *understand* these models with the minimum amount of effort.
2. Then, if the existing vocabularies/ontologies are not able to fully describe the domain in consideration, people *develop* new models. The development process requires interaction with domain experts and end-users in order to produce a model that address the domain under consideration as fully as possible.
3. Finally, people have to *add* data according to the adopted or developed models and to *modify* those data in the future.

Each of these four operations – understanding, developing, adding and modifying – need to be supported by appropriate interfaces that simplify the work of people who may not be expert in ontology-related formalisms and Semantic Web technologies. In this paper, our focus is on the first point of the above list: *ontology understanding*.

Usually, the first activity performed when someone wants to understand the extent of a particular ontology is to consult its human-readable documentation. A large number of ontologies, especially those used in the Linked Data world, have good comprehensive Web pages describing their theoretical backgrounds and the features of their developed entities. However, problems arise when we look at under-developed models, since natural language documentation is usually only published once an ontology has become stable. This approach is justifiable: writing proper documentation costs effort, and re-writing it every time the developing ontology is modified is not practical.

Thus, the only previous way of getting a sense of existing ontologies was to open them in an ontology editor so as to explore their logical axioms. This approach presents practical and cognitive barriers to a person approaching the ontology world for the very first time. First, (s)he has to download and install an ontology editor. Second, (s)he must learn to use the editor, which typically will have a complex and non-intuitive user interface, presenting the user with novel technical terms such as 'axiom' and 'refactor', or common English words used in novel ways, e.g. 'individual' or 'functional'. Then and only then (s)he can try to get a sense of structure of the ontology itself. Obviously these processes are challenging and time-consuming, presenting a barrier that is too great for the majority of non-specialists.

[4] Sindice: http://sindice.com.
[5] Watson: http://watson.kmi.open.ac.uk.
[6] SemNotes: http://smile.deri.ie/projects/semn.

To address this issue, and following the lines of previous works such as *Parrot* [20] and *Neologism* [1], we have developed the *Live OWL Documentation Environment*[7] (*LODE*), an online service that takes any well-formed OWL ontology or, more generally, an RDF vocabulary, and generates a human-readable HTML page designed for browsing and navigation by means of embedded links. In this paper, we introduce this tool, describe its features, and evaluate its usability by considering the success of test users in completing tasks of ontology comprehension.

The rest of the paper is structured as follows. In Section 2 we introduce relevant works in tools for the automatic production of ontology documentation. In Section 3 we present LODE, highlighting its characteristics and features in detail. In Section 4 we present the setting and the details of the test-user experiments we ran to assess its completeness and usability, and in Section 5 we discuss the outcomes of these experiments. Finally, in Section 6, we conclude the paper by sketching out the future developments of our work.

2 Related Works

The production of natural language documentation for ontologies is an important and crucial part of any ontology development process. Such documentation enables users to comprehend the extent of an ontology without having to concern themselves with the particular formal language used to define its axioms. At the same time, writing such documentation manually is an activity that involves a significant effort. Thus, in order to help authors of ontologies to document them, applications have been developed for the automated creation of a first draft of such documentation starting from labels (i.e. *rdfs:label*), comments (i.e. *rdfs:comment*), and other kinds of annotations (e.g. *dc:description*, *dc:creator*, *dc:date*) within the ontology, and from the logical structure of the ontology itself.

SpecGen[8] is a Python tool for the generation of ontology specifications, available as a stand-alone application. It has been used to prepare the HTML documentation of well-known ontologies such as SIOC[9]. SpecGen generates the documentation by processing a pre-defined HTML template into which it adds the list of ontological classes and properties in specific positions. As a result, we obtain a new HTML document where the natural language description of the ontology comes entirely from the template made by authors, while the software takes care of adding the specific information related to the logical structure of the ontology.

In contrast to SpecGen, that needs its base HTML template to work, *VocDoc*[10] is a small Ruby script that allows one to produce documentation starting

[7] LODE, the Live OWL Documentation Environment: http://www.essepuntato.it/lode.

[8] SpecGen: http://forge.morfeo-project.org/wiki_en/index.php/SpecGen.

[9] The Semantically-Interlinked Online Communities (SIOC) project: http://sioc-project.org.

[10] VocDoc: http://kantenwerk.org/vocdoc/.

from RDFS vocabularies and OWL ontologies. It is able to produce both HTML documents and LaTeX files containing the description of the ontology/vocabulary.

Like VocDoc, *OWLDoc*[11] is a fully-automatic generator of a set of HTML pages describing the target ontology. It organises the documentation of each ontological entity in different sections, showing the taxonomy of each entity, the usage of this entity in the context of the ontology, and the formal logical axioms related to the entity (expressed in Manchester Syntax [11]). OWLDoc has been developed as plugin of *Protégé*[12] [12], and as a separate Web application[13].

Oriented to Linked Data applications rather than to ontology documentation, *Paget*[14] is a PHP framework that, upon receipt of an input URL through a browser, dispatches the request according to the particular mime-type specified by the client, and retrieves RDF entities in four different formats: RDF, HTML, Turtle and JSON. It can be used to describe a set of pure RDF statements (*subject-predicate-object*)[15] and, to some extent, to produce a human-comprehensible HTML description from the axioms of an OWL ontology[16].

Neologism[17] [1] is a Web-based editor for the creation of RDFS vocabularies and (very simple) OWL ontologies. Moreover, it implements a publishing system that allows the publication of vocabularies and ontologies on the Web, rendered into natural language HTML pages. Basca *et al.*'s main goal in creating it was to reduce the time needed to create, publish and modify vocabularies for the Semantic Web.

Finally, Parrot[18] [20] is a Web service for the generation of HTML+Javascript documentation of OWL ontologies and RIF rules [3]. This service allows one to specify multiple URLs identifying ontologies in order to produce an HTML summary of them "on the fly", starting from their logical structure and annotations.

3 LODE

LODE, the *Live OWL Documentation Environment*, is a novel online service that automatically generates a human-readable description of an OWL ontology (or, more generally, an RDF vocabulary), taking into account both ontological axioms and annotations, and that orders these with the appearance and functionality of a W3C Recommendation document by use of Cascading Style Sheets (CSS)[19].

[11] OWLDoc: http://code.google.com/p/co-ode-owl-plugins/wiki/OWLDoc.
[12] Protégé: http://protege.stanford.edu/.
[13] Ontology browser: http://code.google.com/p/ontology-browser/.
[14] Paget: http://code.google.com/p/paget.
[15] Ian Davis' Linked Data profile, rendered through Paget: http://iandavis.com/id/me.html.
[16] A vocabulary for describing whisky varieties, rendered through Paget: http://vocab.org/whisky/terms.html.
[17] Neologism: http://neologism.deri.ie.
[18] Parrot: http://ontorule-project.eu/parrot/parrot.
[19] W3C CSS: http://www.w3.org/TR/CSS/.

It automatically extracts classes, object properties, data properties, named individuals, annotation properties, meta-modelling (punning), general axioms, SWRL rules and namespace declarations from any OWL or OWL 2 ontology, and renders them as ordered lists, together with their textual definitions, in a human-readable HTML page designed for easy browsing and navigation by means of embedded links.

LODE is based on an XSLT stylesheet that takes the RDF/XML linearisation of an ontology produced through the OWLAPI[20] [10] as input, and converts it into an HTML representation. If the target ontology is already linearised in that form, it is possible to call the LODE service directly by specifying its URL (i.e. http://www.essepuntato.it/lode/) followed by the complete URL of the ontology, as shown in the following example:

> http://www.essepuntato.it/lode/http://www.essepuntato.it/2008/12/earmark

In the following subsections we introduce the most important features of LODE.

3.1 What axioms are used to create the documentation

Primarily, LODE interprets the most common annotation properties used for the description of entities, in particular[21]: *dc:contributor*, *dc:creator*, *dc:date*, *dc:description*, *dc:rights*, *dc:title*, *dcterms:contributor*, *dcterms:creator*, *dcterms:date*, *dcterms:description*, *dcterms:rights*, *dcterms:title*, *owl:versionInfo*, *rdfs:comment*, *rdfs:isDefinedBy*, *rdfs:label*. LODE adopts the following rules when transforming those annotations in HTML documentation:

- in the presence of Dublin Core annotations defined according to both DC Metadata Elements [9] and DC Metadata Terms [8], the former have precedence;
- dates (i.e. *dc:date* and *dcterms:date*) written according to the XML Schema datatype (i.e. *yyyy-mm-dd*) are automatically transformed into *dd/mm/yyyy*;
- agents (i.e. *dc:creator*, *dc:contributor*, *dcterms:creator* and *dcterms:contributor*) are rendered either as strings or as clickable URLs according to their types, i.e. literals or resources, respectively;
- descriptions (i.e. *dc:description* and *dcterms:description*) are rendered either as strings or as media objects according to their types, i.e. literals or resources, respectively;
- comments (i.e. *rdfs:comment*) and descriptions (i.e. *dc:description* and *dcterms:description*) are represented, respectively, as abstracts and as detailed descriptions of entities;

[20] OWLAPI: http://owlapi.sourceforge.net.

[21] The prefixes *dc*, *dcterms*, *owl* and *rdfs* in the following list respectively refers to "http://purl.org/dc/elements/1.1/", "http://purl.org/dc/terms/", "http://www.w3.org/2002/07/owl#" and "http://www.w3.org/2000/01/rdf-schema#".

– labels (i.e. *rdfs:label*) and QNames (when labels are not specified) are used to refer to all the entities of the ontology, instead of using their URLs;
– the nature of each entity is identified by a descriptive abbreviation, according to its type: “c”, “op”, “dp”, “ap” and “ni” being used to identify class, object property, data property, annotation property and named individual, respectively.

In Fig. 1 and Fig. 2 we show how these annotations are rendered for an example ontology, EARMARK [6], an ontology that defines entities describing document markup (such as elements, attributes, comments and text nodes).

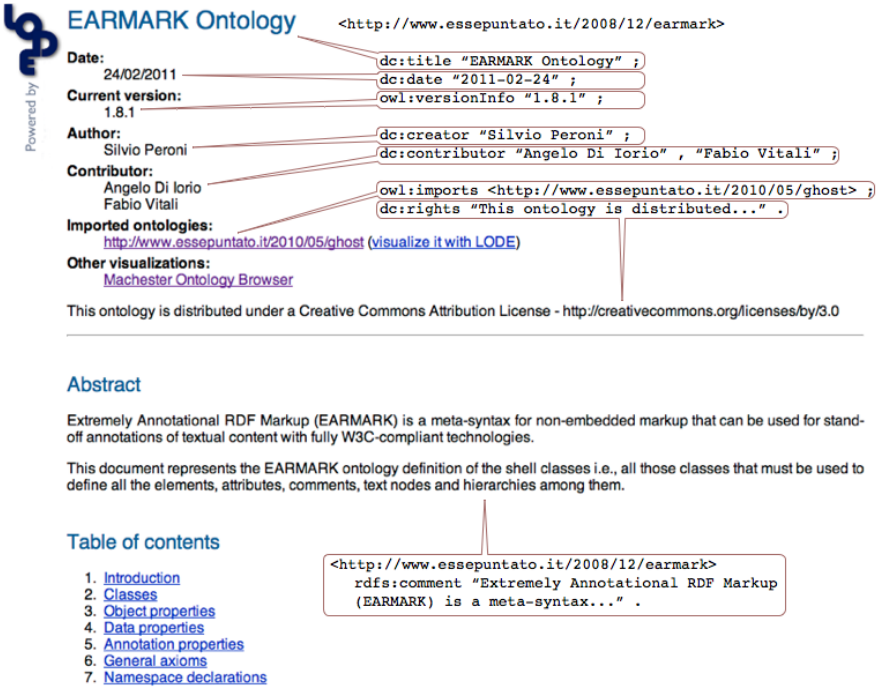

Fig. 1. The beginning of the Web page generated by LODE for the EARMARK Ontology, annotated with OWL assertions in Turtle (not present in the normal LODE web page) illustrating how these assertions are rendered in HTML.

LODE converts all the other axioms of the ontology into Manchester Syntax definitions [11], as shown in Fig. 3. We prefer to use this syntax rather than others since it is the most human-comprehensible syntax for ontological axioms, and thus the most helpful for non-specialists.

Ontological axioms are rendered in grey boxes, one for each entity declared in the ontology. The axioms taken into account by LODE refer to: super-class and super-property, equivalent class and property, disjoint class and property,

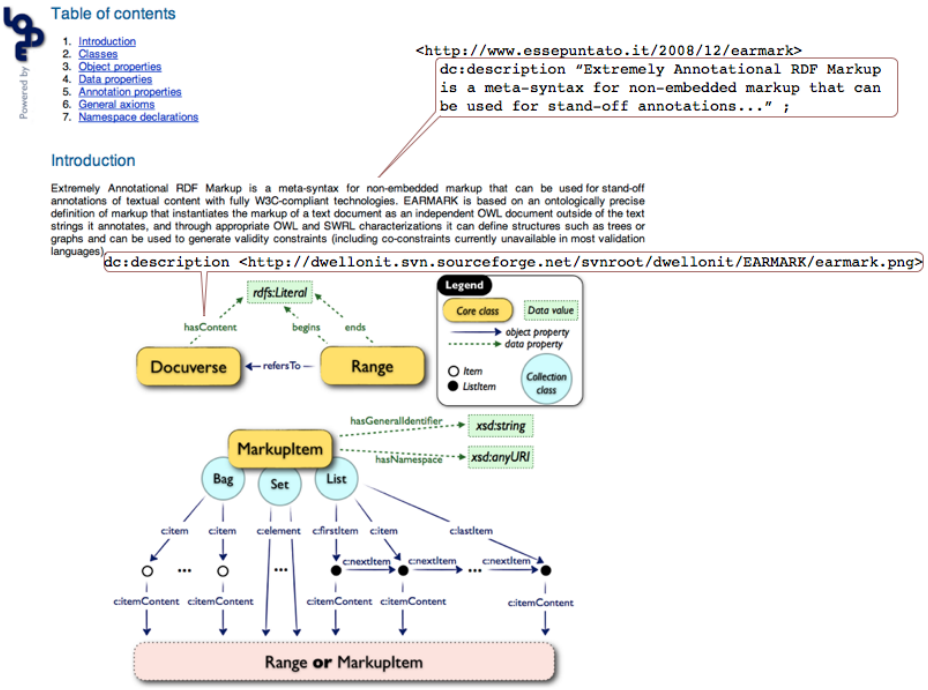

Fig. 2. Two possible kinds of descriptions: pure string (for literals) and media object (for resources).

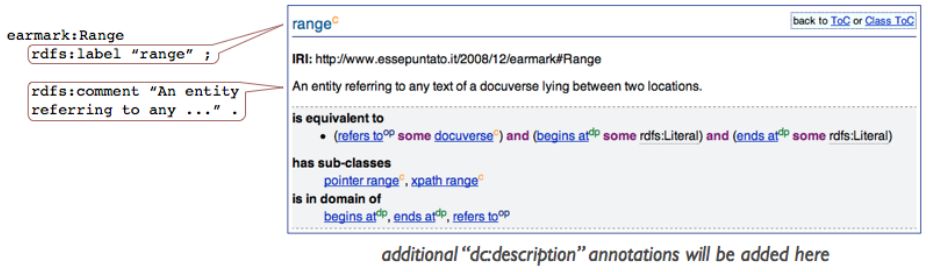

Fig. 3. How entities (classes, properties and individuals) are rendered by LODE.

property domain and range, property chain, keys, object/data property assertion, type, imported ontology, generic axiom and SWRL rule. Moreover, LODE automatically enriches those definitions, adding information about sub-classes, domain/range properties of classes, sub-properties and entity meta-modelling.

3.2 Special parameter usage when calling the LODE service

LODE can be invoked with a number of optional parameters so as to limit or extend the final documentation produced. For instance, it is possible to take into account all the entities in the ontology closure and/or the inferred axioms. The following pseudo-URL describes how to call LODE:

http://www.essepuntato.it/lode/optional-parameters/ontology-url

In particular:

- **www.essepuntato.it/lode** is the URL to call the service;
- **ontology-url** is the full "http://..." URL of the OWL ontology that will be processed by the service. It must be always the last item of the pseudo-URL, and may be preceded by one or more slash-separated parameters.

Fig. 4 illustrates the alternative ways to build the URL to call LODE and the related modules used. The optional slash-separated parameters are described as follows.

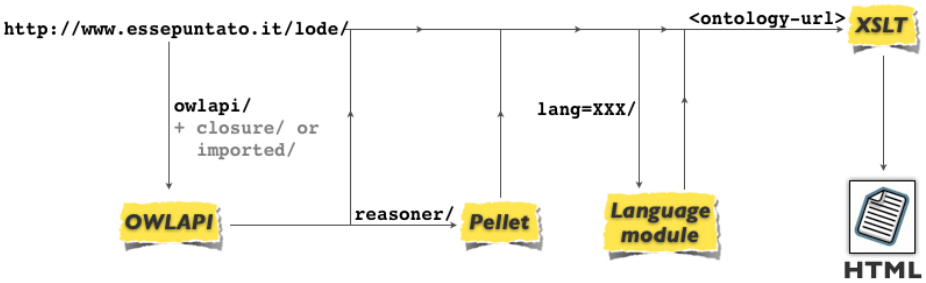

Fig. 4. All the possible ways, according to specific needs, for making a request to LODE.

Parameter "owlapi". When this optional parameter is specified, the ontology defined in *ontology-url* will be pre-processed via the OWLAPI [10], in order to linearise it into the RDF/XML format accepted by LODE. This parameter **is always strongly recommended** to process correctly all those ontologies that were not developed through the OWLAPI.

Parameter "imported". When this optional parameter is specified, the axioms in the imported ontologies of *ontology-url* are added to the HTML description of the ontology itself. This parameter implicitly specifies the *owlapi* parameter.

Parameter "closure". When this optional parameter is specified, the transitive closure given by considering the imported ontologies of *ontology-url* is added to the HTML description of the ontology. This parameter implicitly specifies the *owlapi* parameter. If both the parameters *closure* and *imported* are specified (in any order), *imported* will be preferred.

Parameter "reasoner". When this optional parameter is specified, the inferred axioms of *ontology-url* (through the Pellet reasoner [17]) will be added to the HTML description of the ontology. This parameter implicitly specifies the *owlapi* parameter. Note that, depending on the nature of the ontology to process, this computationally intensive function can be very time-consuming.

Parameter "lang". When this optional parameter is specified, the selected language will be used as the preferred language instead of English when showing the documentation of *ontology-url*. It must be followed by an "=" and the abbreviation of the language to use. E.g. "lang=it" for Italian, "lang=fr" for French, etc. (This presupposes that appropriate language descriptions are present in the ontology.)

3.3 URI fragments

LODE offers intuitive mechanisms to refer to particular ontological entities within the HTML documentation, according to the URL of the entity in consideration. The following extension of the pseudo-URL introduced in Section 3.2 defines how to refer to a particular entity of an ontology:

> http://www.essepuntato.it/lode/optional-parameters/ontology-url **#entity**

For instance, to generate the documentation of EARMARK and then jumping directly to the point where the resource "http://www.essepuntato.it/2008/12/earmark#Element" is described, we need to invoke LODE as follows:

> http://www.essepuntato.it/lode/http://www.essepuntato.it/2008/12/ earmark#http://www.essepuntato.it/2008/12/earmark#Element

This request can be simplified if we look for a description of an entity defined as a *fragment* of the ontology URL, such as the entity *Element* in EARMARK. In this particular case, we can use either the entire entity URL, as illustrated previously, or the entity local name only, as shown as follows:

> http://www.essepuntato.it/lode/http://www.essepuntato.it/2008/12/earmark#Element

3.4 Content negotiation via *.htaccess*

LODE can be used freely by third parties, as described in its documentation. In particular, it may be very usefully employed in conjunction with content negotiation mechanisms to display a human-readable version of an OWL ontology when the user accesses the ontology using a web browser, or to deliver the OWL ontology file itself when the user accesses the ontology using an ontology development tool such as *Protégé* [12] or the *NeOn Toolkit* [19].

LODE can be seen in action by opening, in a Web browser, the EARMARK ontology or any of the SPAR ontologies[22]. For instance, the URL "http://purl.org/spar/fabio" resolves, by content negotiation, to display the LODE HTML version of that ontology with the URL "http://www.essepuntato.it/lode/http://purl.org/spar/fabio". An implementation of such content negotiation is given in [2] by using the *.htaccess* file:

```
AddType application/rdf+xml .rdf
# Rewrite engine setup
RewriteEngine On
# Rewrite rule to serve HTML content
RewriteCond %{HTTP_ACCEPT} !application/rdf\+xml.*(text/html|
    application/xhtml\+xml)
RewriteCond %{HTTP_ACCEPT} text/html [OR]
RewriteCond %{HTTP_ACCEPT} application/xhtml\+xml [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla/.*
RewriteRule ^ontology$ http://www.essepuntato.it/lode/http://
    www.mydomain.com/ontology [R=303,L]
# Rewrite rule to serve RDF/XML content if requested
RewriteCond %{HTTP_ACCEPT} application/rdf\+xml
RewriteRule ^ontology$ ontology.owl [R=303]
# Choose the default response
RewriteRule ^ontology$ ontology.owl [R=303]
```

3.5 Community uptake

LODE is currently used by the following projects:

- The *Semantic Publishing and Referencing* (*SPAR*) project, to generate the documentation for the entire set of SPAR ontologies (available at http://purl.org/spar) created to describe the publishing domain.
- The *VIVO* project[23], an interdisciplinary network enabling collaboration and discovery among scientists across all disciplines, using an open source Semantic Web application originally developed and implemented at the Cornell University. VIVO uses an ontology for data representation and sharing on the Web, documentation for which[24] is generated using LODE.

In addition, LODE has been listed in the *Lower Level Design Tools* page[25] of the official W3C Semantic Web wiki under the category *OWL ontology browsers*, and has been suggested in the Provenance Working Group wiki[26] as one of the documentation tools to use for the *Provenance Ontology* (*PROV-O*)[27].

[22] Semantic Publishing And Referencing (SPAR) ontologies: http://purl.org/spar.
[23] VIVO: http://vivoweb.org.
[24] VIVO OWLAPI Documentation: https://sourceforge.net/apps/mediawiki/vivo/index.php?title=Ontology_OWLAPI_Documentation.
[25] LLDtools: http://www.w3.org/2001/sw/wiki/LLDtools.
[26] Generating HTML documentation of OWL: http://www.w3.org/2011/prov/wiki/Generating_HTML_documentation_of_OWL.
[27] PROV-O: http://www.w3.org/TR/prov-o.

4 Experiments

In order to gather data about the usability of LODE, we undertook user testing. We asked 13 subjects to perform five unsupervised tasks (max. 5 minutes per task), involving ontology navigation through LODE documentation. There were no "administrators" observing the subjects while they were undertaking these tasks. All the subjects were volunteers who responded to personal e-mails or to an invitation sent to the semantic-web@w3.org and public-lod@w3.org lists.

For the tasks, we used a *medium-size* ontology, namely FaBiO, the *FRBR-aligned Bibliographic Ontology*[28], which is composed by 214 classes, 69 object properties, 45 data properties and 15 individuals. FaBiO was chosen because we expected most people involved in the experiments (primarily Semantic Web researchers and practitioners) to have familiarity with the domain it describes, i.e. bibliographic entities such as research papers, journal articles and books. In addition, FaBiO was also chosen because using an ontology larger than FaBiO would have required more time to complete the tasks, potentially reducing the number of users willing to complete the evaluation, and thus reducing the number of useful data for the evaluation.

The tasks given to the subjects are shown in Table 1. This set of tasks was designed to exploring the LODE capabilities in creating a human-readable documentation and in browsing ontologies.

Table 1. The five tasks subjects performed in the user testing session.

Task 1	Describe the main aim of the ontology.
Task 2	Describe what the class *doctoral thesis* defines.
Task 3	Describe what the object property *has subject term* describes, and record its domain and range classes.
Task 4	Record the class having the largest number of direct individuals (i.e. individuals that belongs explicitly to that class and that are not inferable from its subclasses)
Task 5	Record all the subclasses and properties involving the class *item*

Task 1 is a pure descriptive activity that involves only the documentation produced by LODE, without using any navigational features such as Web links. Task 2 and 3 are similar to Task 1, but in addition typically requires the user to use some navigational facilities to reach the class *doctoral thesis* and the object property *has subject term*. Finally, Tasks 4 and 5 further assess how easily LODE enables users to browse the ontology and understand its structure. Our interest was to assess how well LODE helped users by producing documentation of an OWL ontology, enabling them more easily to browse and make sense of it.

[28] FaBiO ontology: http://purl.org/spar/fabio.

The test session was structured as follows. We first asked subjects to complete a short multiple-choice questionnaire about their background knowledge and skills in OWL, ontology engineering and ontology documentation (max. 2 minutes). Then, as a warm-up task, we asked subjects to use LODE to explore the FOAF ontology[29], a relatively simple ontology, in order to become familiar with the structure of the documentation it produced, and with its navigation mechanisms (primarily, internal hypertext links) (max. 5 minutes). Then, as the real test, we asked subjects to complete the five tasks listed in Table 1 using the documentation of the FaBiO ontology created by LODE (ideally 2 minutes, max. 5 minutes, per task). Finally, we asked subjects to fill in two short questionnaires, one multiple choice and the other textual, to report their experience of using LODE to complete these tasks (max. 5 minutes). All the questionaries and all the outcomes of the experiments are available online[30].

5 Evaluation

Out of 65 tasks in total (5 tasks given to each of 13 subjects), 58 were completed successfully (i.e., the right answers were given), while 7 had incorrect answers or were not completed at all, giving an overall success rate of 89%. The 58 successes were distributed as follows: 13 (out of 13) in Task 1, 13 in Task 2, 13 in Task 3, 10 in Task 4 and 9 in Task 5.

The usability score for LODE was computed using the *System Usability Scale* (*SUS*) [4], a well-known questionnaire used for the perception of the usability of a system. It has the advantage of being technology independent (it has been tested on hardware, software, Web sites, etc.) and it is reliable even with a very small sample size [16]. In addition to the main SUS scale, we also were interested in examining the sub-scales of pure *Usability* and pure *Learnability* of the system, as proposed recently by Lewis and Sauro [13]. As shown in Table 2, the mean SUS score for LODE was 77.69 (in a 0 to 100 range), abundantly surpassing the target score of 68 to demonstrate a good level of usability [16]. The mean values for the SUS sub-scales Usability and Learnability were 76.4 and 82.7 respectively. In addition, two sub-scores were calculated for each subject by considering the values of the answers given in the background questionnaire, composed of ten questions about the subject's experience with ontologies and two questions about his/her experience with ontology documentation tools. We compared these sub-scores with the SUS values and the other sub-scales using the Pearson's r. We found a small negative correlation (between -0.34 and -0.14) between the experience sub-scores and the SUS values. This may show that the perceived usability of LODE does not depend upon any particular ability of subjects in the use of ontologies and ontology documentation tools, rather the opposite. However, each correlation measure appears to be not statistically significant and we need to enrich our dataset to come to a more precise conclusion.

[29] FOAF ontology: http://xmlns.com/foaf/spec/index.rdf.
[30] http://www.essepuntato.it/2012/04/lodeusertesting.

Table 2. System Usability Scale values and related sub-measures.

Measure	Mean	Max. value	Min. value	Standard deviation
SUS value	77.7	92.5	57.5	12.5
Usability	76.4	90.6	56.3	12.8
Learnability	82.7	100	62.5	14.9

Axial coding of the personal comments expressed in the final questionnaires [18] revealed a small number of widely perceived issues. Only 11 out of the 13 subjects tested had meaningful comments that were used for the study, and, of the 15 terms that were identified as significant in the comments, only five (three positive and two negative) were mentioned by more than one individual (albeit sometimes with different words), as shown in Table 3.

6 Conclusions

Writing good documentation for ontologies manually is costly in effort, and is impracticable for an ontology that is under active development. In this paper we addressed this issue by introducing LODE, the *Live OWL Documentation Environment*, a Web application that automatically creates human-readable HTML ontology documentation on-the-fly from an OWL file of ontological axioms and annotations. We discussed how it can be used in conjunction with content negotiation that delivers LODE documentation to a browser window or the OWL file itself to an ontology editor. And we assessed LODE's usability and effectiveness in creating browsable ontology documentation by conducting user testing. The results of these tests are encouraging, demonstrating that LODE provides a stable service for the automatic creation of useful ontology documentation.

Some days before the deadline for the camera ready version of this paper, we performed another user testing session that compared LODE with similar tools for the automatic generation of ontology documentation, namely Parrot [20] and the OWLDoc-based Ontology Browser. A preliminary and informal evaluation of the outcomes of this new testing session confirmed the usability results highlighted in this paper. We also performed a comparison of the performances of the tools with users trying to carry out the tasks in Table 1. Early outcomes seem to confirm the usefulness of LODE when dealing with such tasks. This work will be reported at a later date. In future, we plan to extend LODE features to include suggestions highlighted by our users. In addition, we plan to conduct another user testing session to compare LODE with other ontology visualisation and browsing tools such as KC-Viz [14] and OWLViz [12].

Acknowledgements. Aspects of this work have been supported by the *JISC* (*Joint Information Systems Committee*) through grant support to DS. We thank all the people who took part to the user testing session.

Table 3. Terms – three positive (+) and two negative (-) – mentioned by more than one individual in the final questionnaire responses.

Term	Description	Frequency
Search (-)	No search function was provided to directly look for and access entities of the ontology. Users acknowledge that since the ontology is on a single web page, they could use (and in fact did use) the search function of the browser, but many still found it a missing feature	7 out of 11
Readability (+)	High praise was given to the clarity of the presentation, the intuitiveness of the organization, and the immediacy of identifying the sought information. The good typographical style of the output is clearly among the best qualities of LODE	5 out of 11
Links within the document (+)	The systematic use of internal links to the various features of the ontology was considered useful and immediately usable	4 out of 11
Scalability (-)	The LODE interface provides no links to entities provided by external, but linked ontologies. A highly modular ontology composed of a multiplicity of independent sub-ontologies is hard to navigate, and similarly the structure on a single page could make really large ontologies quite hard to access	3 out of 11
Single page (+)	Praises were given to the idea of placing all the content on a single Web page, which allows a multiplicity of approaches to accessing and reading the ontology, including visual transitions and scrolling that would not be possible if the ontologies had been presented in separate web pages.	2 out of 11

References

1. Basca, C., Corlosquet, S., Cyganiak, R., Fernández, S., Schandl, T. (2008). Neologism: Easy Vocabulary Publishing. In Proceedings of the 4th Workshop on Scripting for the Semantic Web. http://ceur-ws.org/Vol-368/paper10.pdf (last visited June 28, 2012)
2. Berrueta, D., Phipps, J. (2008). Best Practice Recipes for Publishing RDF Vocabularies. W3C Working Group Note, 28 August 2008. World Wide Web Consortium. http://www.w3.org/TR/swbp-vocab-pub/ (last visited June 28, 2012)
3. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D. (2010). RIF Core Dialect. W3C Recommendation, 22 June 2010. World Wide Web Consortium. http://www.w3.org/TR/rif-core/ (last visited June 28, 2012)
4. Brooke, J. (1996). SUS: a "quick and dirty" usability scale. In Usability Evaluation in Industry: 189-194. London, UK: Taylor and Francis. ISBN: 978-0748404600

5. d'Aquin, M., Motta, E. (2011). Watson, more than a Semantic Web search engine. In Semantic Web – Interoperability, Usability, Applicability, 2 (1): 55-63. DOI: 10.3233/SW-2011-0031
6. Di Iorio, A., Peroni, S., Vitali, F. (2011). A Semantic Web Approach To Everyday Overlapping Markup. In Journal of the American Society for Information Science and Technology, 62 (9): 1696-1716. DOI: 10.1002/asi.21591
7. Dragan, L, Handschuh, S., Decker, S. (2011). The semantic desktop at work: interlinking notes. In Proceedings the 7th International Conference on Semantic Systems (I-SEMANTICS 2011). DOI: 10.1145/2063518.2063521
8. Dublin Core Metadata Initiative (2010). DCMI Metadata Terms. DCMI Recommendation. http://dublincore.org/documents/dcmi-terms/ (last visited June 28, 2012)
9. Dublin Core Metadata Initiative (2010). Dublin Core Metadata Element Set, Version 1.1. DCMI Recommendation. http://dublincore.org/documents/dces/ (last visited June 28, 2012)
10. Horridge, M., Bechhofer, S. (2011). The OWL API: A Java API for OWL ontologies. In Semantic Web – Interoperability, Usability, Applicability, 2 (1): 11-21. DOI: 10.3233/SW-2011-0025
11. Horridge, M., Patel-Schneider, P. (2009). OWL 2 Web Ontology Language: Manchester Syntax. W3C Working Group Note, 27 October 2009. World Wide Web Consortium. http://www.w3.org/TR/owl2-manchester-syntax/ (last visited June 28, 2012)
12. Knublauch, H., Horridge, M., Musen, M. A., Rector, A. L., Stevens, R., Drummond, N., Lord, P. W., Noy, N. F., Seidenberg, J., Wang, H. (2005). The Protege OWL Experience. In Proceedings of the OWLED 05 Workshop on OWL: Experiences and Directions. http://ceur-ws.org/Vol-188/sub14.pdf (last visited June 28, 2012)
13. Lewis, J. R., Sauro, J. (2009). The Factor Structure of the System Usability Scale. In Proceedings of the 1st International Conference on Human Centered Design (HCD09). DOI: 10.1007/978-3-642-02806-9_12
14. Motta, E., Mulholland, P., Peroni, S., dAquin, M., Gomez-Perez, J. M., Mendez, V., Zablith, F. (2011). A Novel Approach to Visualizing and Navigating Ontologies. In Proceedings of the 10th International Semantic Web Conference (ISWC 2011). DOI: 10.1007/978-3-642-25073-6_30
15. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. In International Journal of Metadata, Semantics and Ontologies, 3 (1): 37-52. DOI: 10.1504/IJMSO.2008.021204
16. Sauro, J. (2011). A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices. ISBN: 978-1461062707
17. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. In Journal of Web Semantics, 5 (2): 51-53. DOI: 10.1016/j.websem.2007.03.004
18. Strauss, A. Corbin, J. (1998). Basics of Qualitative Research Techniques and Procedures for Developing Grounded Theory (2nd edition). Sage Publications: London. ISBN: 978-0803959408
19. Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (2012). Ontology Engineering in a Networked World. Berlin, Germany: Springer. ISBN: 978-3642247934
20. Tejo-Alonso, C., Berrueta, D., Polo, L., Fernandez, S. (2011). Metadata for Web Ontologies and Rules: Current Practices and Perspectives. In Proceeding of the 5th Interational Conference on Metadata and Semantic Research (MTSR 2011). DOI: 10.1007/978-3-642-24731-6_6