Getting it Fast
An Information Systems Oriented Semantic Web Curriculum

Daniela Giordano and Francesco Maiorana
Department of Electrical, Electronic and Computer Engineering, University of Catania, Viale A. Doria, 6, Catania, Italy

Keywords: Information Systems Applications, Mash-ups, RDF, RDS, OWL, Linked Open Data, Knowledge Representation and Reasoning, Ontology Engineering, Triplify, SPARQL, Gruff, Jena.

Abstract: The fast growing importance of the semantic web and semantic web applications is demonstrated by the exponentially growing amount of semantic data produced on the web and by the rise of the linked open data movement. Besides this strong interest there is a request both from academia and industry for well-prepared students in order to minimize the training effort needed to prepare them for productive work. This paper describes the teaching experience of a Master's course entitled "Laboratory in software design and development – semantic technologies". A detailed description of the curriculum, the rationale underlying the choice of content and the software tools used, as well as the main lessons learnt from the experience, are presented.

1 INTRODUCTION

The enormous impact of the World Wide Web (WWW) is hindered by some problems related to:

- data representation: all the data on the WWW are semantics-free. This gives rise to disaligned, inconsistent and unrelated information.
- entity update: an update on an entity does not affect all the pages that semantically reference it.

The semantic web is able to overcome these difficulties and is gaining momentum day by day. The rise of the semantic web has created a strong demand by industry for people trained in semantic technologies as well as ontologies. This demand is estimated to quickly grow in the following years. Paralleling this strong interest there is a flourishing growth of courses in academia and professional institutions; however, these courses are often single courses within larger programs. The authors in (Neuhaus, 2011) identified only one complete academic program completely devoted to education in applied ontology (a Master's program at the University of Buffalo) and 21 programs that offer ontology-centered topics.

Applied ontology and the semantic web are relatively young disciplines and therefore there is not yet a general consensus on questions related to several aspects ranging from the curriculum to even the terminology.

In the ACM/IEEE (ACM/IEEE, 2013) computer science curriculum, for example, the knowledge required for an ontologist is spread among courses or topics within courses such as the elective "Advanced Representation and Reasoning" in the field of Intelligent Systems (IS), dealing with ontology engineering and, in the Computational Science field, the elective course entitled "Data, Information and Knowledge", dealing with "Knowledge: ontologies, triple stores, semantic networks, rules", and so on. First, this paper will briefly review teaching literature, curriculum examples and didactic tools in the main areas of the course: ontology design, ontology query languages, ontology programming framework, mash-up and semantic web applications. The design of an introductory lab in semantic web technologies, worth 3 ECTS, will also be presented.

Among the several collections of semantic web educational resources to which the reader can refer (Diederich, 2007), there exists a repository built by the European Association for Semantic Web Education (Diederich, 2006). In spite of this field being relatively young, there are outstanding experiences in teaching ontology engineering and ontology design. In (Rewctor, 2004) the authors report the results of a several-years-long experience

Giordano D. and Maiorana F..
Getting it Fast - An Information Systems Oriented Semantic Web Curriculum.
DOI: 10.5220/0004848300420048
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 42-48
ISBN: 978-989-758-021-5

in teaching the OWL language and ontology design, along with common errors and common design patterns. Several ontologies have been used as didactic examples to teach ontology design and OWL, such as Pizza, Wines and Marsupials.

Recently, a didactic example in the biological domain was proposed by (Schober, 2012), where the authors propose an OWL ontology describing zoo animals. The ontology, with respect to other large biomedical ontologies, has the advantage of massively reducing the number of classes and overall complexity, allowing for fast memorization and orientation, yet it uses all the major ontology constructs, with related modelling challenges together with some Ontology Design Pattern (ODP). The proposed ontology uses a light version of Bitop (Beißwanger, 2008) as a top level ontology. Moreover, the ontology covers a domain that involves common knowledge and therefore can be easily understood by the majority of people, especially in the life science domain. The ontology was used in a complete curriculum for ontology design described in (Boeker, 2012) where the authors present a teaching experience of a one-week workshop organized around 16 modules, addressing 4 main themes with an increasing level of design complexity: basic principles, practical ontology design, using top-level ontologies and ODP.

Various works address the development of software tools used in training students, training and teaching ontology query languages, and examples of clear and didactically sound tutorials are also available. A recent example of a software training tool is described in (Gerber, 2010) where the authors present a web-based SPARQL trainer that allows the tutor to design a course along a set of concepts that are to be tested. The tutor prepares a set of questions and a dataset. Each question is provided with a query solution against which the student solution is compared. The comparison is based on the result sets: the query solution and the student solution result set must coincide both in the elements and in their order.

For mash-up and semantic web applications, a good tutorial can be found in (Della Valle, 2008) where the author presents a semantic web application that expects a music style as an input, retrieves data from online music archives and event databases, merges them by a bridge ontology and allows the user to explore events related to artists that practice the required style.

The aim of this paper is to give a detailed overview of the curriculum taught in a Master's-level class in Informatics Engineering in order to "describe in more detail" how some of the knowledge and skills indicated in (Neuhaus, 2011) can be acquired. Our curriculum is "IT-oriented" with a particular emphasis placed on the knowledge and deployment of "IT systems involving many components in addition to the ontology itself" (Neuhaus, 2011). This type of curriculum fits well with the knowledge and background of Informatics Engineering and Computer Science students and provides a means of broadening both data and knowledge design capabilities, as well as system integration and software development skills.

The paper is organized as follows: section 2 presents the teaching context and the contents; section 3 describes the lessons learnt, highlighting strengths and weaknesses of the approach used in the course; section 4 reports conclusions and describes further work.

2 TEACHING CONTEXT AND CONTENTS

The teaching experience pertains to a 30-hour course within the Informatics Engineering Master's degree at the University of Catania. The course was taught in the 2011 Fall term. The course was an elective one, leading to a final certification. It was attended by both first year and second year students. Management of the class was not particularly easy due to the students' varying levels of knowledge. The course schedule was a three hour meeting per week for ten weeks. The course was designed around semantic technologies, with the aim of demonstrating during the classes the main technologies of the semantic web and the main tools used to design and implement semantic applications. As a university rule, the course was graded on a pass/fail basis.

The students attending the course could get acquainted with the following topics and questions:

1) Data transformation from a relational database to RDF. After a review of the main tools, Triplify (http://triplify.org/Overview) was chosen due to its simplicity and its tight relationship with database technologies and the PHP web programming language, tools that are well-known to all the students attending the course.

2) Querying triple stores/RDF files. The main tool used was the SPARQL query language. The Gruff browser was used both in a standalone version and with the AllegroGraph server. Among the different advantages of the Gruff browser, the graph

representation capabilities were considered a definitive plus.

3) Relations and ontology design using RDFS and OWL. Protégé was the tool chosen, due to its flexibility, integration with reasoners, rich availability of plugins and its free availability.

4) Linked Open Data: design and implementation of a mash-up from linked data. The Application Framework, Jena, was chosen.

The course also offered students possibilities to familiarize themselves with the following core skills and knowledge:

- Clarifying the purpose of a given ontology
- Judging what kinds of ontologies are useful for a given problem
- Identifying, evaluating and using software tools that support ontology development
- Using (reading and writing) different representation languages
- Conducting ontological analyses
- Using a modern programming language
- Working in teams

The class activities were supported by an on-line class site developed with the Moodle platform, used for sharing lecture notes, projects and communications and greatly simplified the management of class activities.

The class schedule is reported in Table 1 with an indication of the time required for each topic.

The table gives an overview of the curriculum content that was organized into 4 main parts or modules, with an increasing level of difficulty and with a broader view of the covered topics. Each module was further divided into 2 or 3 units. The modularization of the curriculum allows for an easier customization for other courses and classes with different backgrounds.

The first module presents the Semantic Web in comparison to the WWW and its main contributions. The RDF language and RDF graph are presented along with the major serialization languages: N-triples, turtle and XML. Concepts like reification, blank node and list and their impact in designing are presented and applied to examples and exercises. An introduction to the main ontological languages RDFS, RDFS-Plus, OWL and their expressiveness is undertaken.

The second unit deals with data transformation and query techniques. The data transformation technique was appreciated by the students and served as a common thread between their strong background on database and web programming and the course content. After a review of the main tools, such as D2RQ, Triplify was chosen due to its simplicity and its affinity with SQL and the PHP development environment. This allowed students to publish RDF triples from their own databases as well as from data present in Bulletin Board, Content Management systems and so on. The students

Table 1: Class schedule.

Topic	#H	Tools	Readings	Assignment
Part 1: Basic Principles				
The Semantic web	3		(Allemang, 2011) Ch. 1	
RDF and RDF graph. Serialization languages.	3		(Allemang, 2011) Ch. 2-3	
Reification, blank node, list in RDF. RDFS, RDFS-Plus, OWL	3		(Allemang, 2011) Ch. 3-4	
Part 2: Data transformation and querying				
Data transformatio n tools. Lab session: use of Triplify	3	Triplify Allegro-graph	Auer, 2009	Triplify a Database
SPARQL and SPARQL 1.1	4	Allegro-graph/ Gruff	Ch. 5 (Allemang, 2011), Ch. 6 (Liyang, 2011), (Bizer, 2009)	Design a set of queries and evaluate their performance
Part 3: Ontology design				
Ontology engineering. Use of Protegè	4	Protegè/ OWLViz	Ch. 4 (Della Valle, 2009), Ch. 6, 7 8, 9, 11, 12 (Allemang, 2011) ch. 4, 5, 12 (Liyang, 2011)	Design an ontology
The Jena framework	4	Eclips/ Jena/ MYSQL	Ch. 12, 13, 14, 15 (Liyang, 2011), (McBride, 2002), (Carroll, 2004)	Develop an application that integrates data from FOAF and DBpedia
Part 4: Semantic web applications: mash-up, linked data and visualization				
Visualization framework: Exhibit	2	Exhibit	Huynh, 2007	Visualize events data on google maps
A semantic web application: mash-up and linked data	4	Combi-nation of tools	(Bizer, 2009 (b)) and (Heath, 2011), Della Valle, 2008) Ch. (Della Valle, 2009)	Develop a mash-up application (optional)
Total	30			

became acquainted with large RDF files and began to design RDF graphs starting from an E/R schema.

The original work by (Auer, 2009) describing the design of Triplify was used as a main reference. The best works were presented by the students, which benefited the entire class. The use of a shared pool of projects, (Giordano, 2004) can be viable, even if course is in its first edition, by using the projects developed throughout the year, to facilitate the students in overcoming difficulties, finding technical details and solutions as well as to gain inspiration from the work of their peers.

The subsequent step was the introduction of the SPARQL language to query a RDF triple store. The SPARQL language was selected since it resembles the well-known database query language SQL. The use of triples, graph patterns and other features allowed the students to practice with the underlying RDF graph and with serialization languages. Moreover, the students were free to use OWL and domain ontologies and to practice with large triple stores such as DBpedia. For all these reasons the introduction of SPARQL was considered a beneficial step in this phase of the course. The students were free to choose the RDF store where they published their data. The suggested choice was the AllegroGraph RDF store (http://www.franz.com/agraph/allegrograph/) and the Gruff graph based triple store browser (http://www.franz.com/agraph/gruff/) due to the graph representation of the results set. Other common choices were the Sesame (http://www.openrdf.org/) and Joseky (http://joseki.sourceforge.net/) RDF stores.

Chapter 5 of (Allemang, 2011) and chapter 6 of (Liyang, 2011) were used as main references. All the types of queries were covered. The unit started with the select query form presenting triple and graph patterns, query modifiers, optional patterns that could be nested, filter conditions on different types and regular expressions, union of graph patterns, background and named graphs to query multiple graphs. The other types of query forms, namely construct and rules, describe and ask, were then covered followed by the new SPARQL 1.1 features such as aggregate functions, group by, sample and bound, subqueries, negation, expressions with SELECT, property paths, transitive queries, federated queries and the SPARQL update 1.1 standard. In order to make the students aware of performance issues in query formulation the problem of benchmarking was presented through studies such as (Bizer, 2009) and (Schmidt, 2009). These works were also used as query examples. Tools such as (http://ftp.heanet.ie/disk1/download.sourceforge.net/pub/sourceforge/b/project/bs/bsbmtools/bsbmtools/bsbmtools-0.2/) were used by the students to profile queries and experiment with different query formulations. The students experimented with different datasets such as the one suggested in (Bizer, 2009), FOAF RDF files, well-known public domain ontologies such as the Wine ontology and so on. Figure 1 shows an example of a graph created with the Gruff browser and Table 2 shows a sample output of the bsbmtools available in the Berlin SPARQL Benchmark.

The third part of the course was divided among two related units. The first dealt with ontology design principles. The main references were chapter 4 of (Della Valle, 2009) as well as chapters 6, 7 for RDFS design and chapters 8, 9, 11, 12 from (Allemang, 2011) for OWL-related design. Finally, chapter 4 for RDFS design, chapter 5 for OWL design and chapter 12 for tools and resources from (Liyang, 2011) were also suggested to the students for reference. Suggestions on how to present the material are reported in the Lessons Learnt section. As a working tool, Protégé was chosen (http://protege.stanford.edu/) with the OWLViz (http://www.co-ode.org/downloads/owlviz/) for class visualization. The approach was to follow the Protegè tutorial (Horridge, 2009) as well as sharing suggestions such as those reported in (Rewctor, 2004) and (Noy, 2004), just to cite some examples. An initial methodology and a practical example for converting an E/R model into an OWL model was presented, along with some ODP. Other literature was suggested as further readings, i.e., (Poveda-Villalón, 2010) and (Hammar, 2010). Two practical ontology design exercises were also proposed, the first one about the design of an educational portfolio and the second one regarding the sharing of learning objects. As a framework to develop semantic applications and manage ontologies, the Jena framework was chosen due to the large use of Java technologies in both academia and the professional world, as well as its free availability.

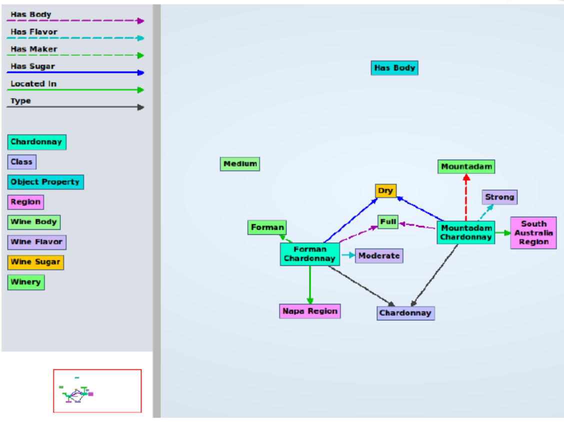
The Eclipse IDE and the MYSQL database to store persistent models, were also suggested for data manipulation in a database. As a reference for practical applications, chapters 12, 13, 14 and 15 of (Liyang, 2011) were suggested.

To present the Jena framework and its main functionality and usage, the guidelines of presentation indicated in (McBride, 2002) and (Carroll, 2004) were followed. Following this in-depth introduction on the architecture and main features, practical examples on creating, using, manipulating and navigating through RDF graph, reading and storing to file as well as to database were given with practical applications and working code. Practical examples performing queries along with using reasoners were also presented. The tutorial used in the previous model was recreated using, programmatically, the OWL API. Finally, a complete example of a linked data application in the style of (Liyang, 2011) was illustrated. The example was used as a starting point for a linked data application. The practical approach to linked data was followed by a theoretical overview. The books (Bizer, 2009 (b)) and (Heath, 2011) were used as reference material and an overview of the main concepts, design principles, techniques and recipes for publishing and consuming linked data was presented to the students.

```
PREFIX vin: <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine#>

SELECT * WHERE {
 ?winery a vin:Chardonnay .
 MINUS{?winery vin:hasBody vin:Medium .}
 ?winery ?property ?value
}
```

(a)



(b)

Figure 1: An example of a SPARQL 1.1 query (a) with the graph results visualized with the Gruff browser (b).

Finally, the fourth part completed the picture by presenting visualization tools and strategies to design and implement a mash-up semantic application. Exhibit was selected as the visualization tool due to is simplicity and flexibility in addition to its free availability (Huynh, 2007) together with tutorials and other learning material. In particular, Exhibit 3.0 was presented to the students, together with its use in a local environment.

In the original plan, all the technologies tools and techniques had to converge in the presentation of the final project. The aim was to present a mash-up application that integrates different sources of data in different formats, (e.g., relational, XML and RDF) by designing a bridge ontology integrating all these data in a unifying view. The reference material was the tutorial (Della Valle, 2008) as well as chapter 9 of (Della Valle, 2009) where the design principles and main techniques and implementation details are presented. This kind of application should be considered as the concluding and unifying aspect of the course, where all the principles and technologies can be applied to build a working semantic application.

Table 2: Result of the performance profile of the query on the dataset described in (Schmidt, 2009) with 1.000 as scale factor. The scale factor used is related to the number of products.

Query: Find information of the Person whose label is "reexhibit"
describe ?x
where {
?x rdfs:label "reexhibit".
}

Output of the bsbmtools-0.2 Scale factor: 1000 Number of warmup runs: 20 Seed: 808080 Number of query mix runs (without warmups): 50 times min/max Querymix runtime: 2.6075s / 3.1330s Total runtime: 140.452 seconds QMpH: 1281.58 query mixes per hour CQET: 2.80904 seconds average runtime of query mix CQET (geom.): 2.80503 seconds geometric mean runtime of query mix Metrics for Query: 1 Count: 50 times executed in whole run AQET: 0.004793 seconds (arithmetic mean) AQET(geom.): 0.004764 seconds (geometric mean) QPS: 208.64 Queries per second minQET/maxQET: 0.00429478s / 0.00822679s Average result (Bytes): 1320.00 min/max result (Bytes): 1320 / 1320 Number of timeouts: 0

The guiding factors for choosing the tools were: the efficiency, the ease of use and its connection with the database technologies for Triplify, the visualization aids for Gruff, the free and open source availability together with its large use in ontology development for Jena, and the relative ease of use and rich set of graphical features for Exhibit.

Overall there was a positive appreciation of the course, particularly from the more motivated students. Some of them expanded upon the course content with a Master's thesis.

3 LESSONS LEARNT

Analyzing the teaching experience, it is possible to make the following considerations. The initial module on the semantic web, although an interesting and necessary step that gives an overview of the potentiality and breadth of the fields, can be reduced to a minimum and allow the important concepts to emerge from practical case studies such as querying an RDF graph and ontology design.

The use of Triplify was a good ice breaking activity allowing students to start from what they already knew well, namely database design and web programming, bringing them to fast production of their RDF data and hence reading and navigating the data produced.

The use of SPARQL for querying RDF store was a definitive plus, serving as a bridge between different concepts: it allows the student to master serialization languages, ontology representation and navigation, triple and graph pattern, query language, and so on. The use of a benchmarking tool can be of great benefit by allowing the students to place more attention on query design and underlying execution mechanisms. This type of knowledge can help in obtaining a better understaning of both the querying and, more broadly, of the ontology design processes.

The ontology design step is worth further consideration. This can be done by allowing more time for student projects and by sharing both their projects and the peer reviews along with the trainer's feedback and comments (Giordano, 2004). The use of conversion from E/R to OWL is a good starting point. Supplementing ODP by applying it to a real case scenario could be of great benefit. The amount of material suggested to the student should be covered inside the development of a practical project in order to avoid an overwhelming effect.

The programming side of the linked data and mash-up applications can be the central aspect of courses for Informatics Engineering and Computer Science students. By leveraging their strong background on software design and programming, by scheduling the necessary time and granting a valid number of credit units, there is the possibility to develop, through group projects, interesting examples of applications.

Some students reported the preference to begin with ontology design and then view the querying aspects. The best curriculum may then be designed around a parallel development of query, ontology design and programming framework to converge as soon as possible with the design and implementation of linked data and mash-up in a semantic web application.

Due to the breadth and richness of the field along with the cognitive load needed for mastering all the topics and theoretical concepts as well as the set of technical expertise, both as an IS engineer and as designer and programmer, a course with more than 30 hours is required. In practice, due to an initial underestimation of the overall workload, the requirement for the final project was turned into an optional one (only 10% of the students completed it) and the course assessment was performed based on three comprehensive exercises that demonstrated a working knowledge of the techniques and tools showcased during the course. With a course of 40 to 60 hours, more time could be devoted to practical projects with a wider scope, to be developed both in lab sessions, under the guidance of a tutor, and at home, leaving the contents as described. Some informal feedback collected by students who entered the workforce after graduating and were assigned responsibilities involving semantic web technologies, commented that although the contents were covered in a very condensed timespan, they provided an effective groundwork.

4 CONCLUSIONS AND FURTHER WORK

This paper has presented a curriculum, oriented towards Information Technology, suited for a course on semantic web and ontologies. The syllabus is organized around 4 major modules allowing for easy customization. The curriculum, by leveraging on the database, software engineering and web programming backgrounds of the participants, lays the foundation for mastering semantic web theoretical essentials, as well as the tools and techniques necessary to develop applications that exploit the full potential of this emerging field.

As further study a systematic analysis of the mini-projects handed in by the students is planned, to document common errors and pitfalls, for example, in the query formulation with respect to both the desired result and the execution efficiency, to adjust accordingly further versions of this elective course.

REFERENCES

Allemang, D., Hendler, J., 2011. *Semantic web for the working ontologist. Effective Modelingin RDFS and*

OWL. Second Edition.(2011) Morgan Kaufmann.
Association for Computing Machinery, IEEE-Computer Society, 2012. Draft Computer Science Curricula 2013.
Auer, S., S. Dietzold, Lehmann, J., Hellmann, S., Aumueller, D., 2009. Triplify - Light-weight Linked Data publication from relational databases. Proceedings of the *World Wide Web Conference* (WWW) (2009), pp. 621-630.
Beißwanger, E et al., 2008. BioTop: An Upper Domain Ontology for the Life Sciences - A Description of its Current Structure, Contents, and Interfaces to OBO Ontologies, In: *Applied Ontology* 3:4, pp. 205- 212.
Bizer, C.,Schultz, A., 2009. The Berlin SPARQL benchmark. *International Journal on Semantic Web and Information Systems* 5(2): pp. 1-24.
Bizer, C., Heath, T., Berners-Lee. T., (2009 b) "Linked data-the story so far." *International Journal on Semantic Web and Information Systems (IJSWIS)* 5.3 (2009): 1-22.
Boeker, M , Schober, D., Raufie, D.,Grewe, N., Rohl, J., Jansen, L., Schulz, S., 2012. Teaching good biomedical ontology design. In Cornet R & Stevens R (eds.), Proceedings of the *3rd International Conference on Biomedical Ontology* (ICBO 2012), KR-MED Series, July 21-25, 2012, Graz, Austria. URL http://ceur-ws.org/Vol-897/sessionJ-paper25.pdf.
Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K., 2004. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters,* pp. 74-83. ACM.
Della Valle, E., Celino, I., Cerizza, D., 2009. *Semantic Web: dai fondamenti alla realizzazione di un'applicazione*. Pearson – Addison Wesley.
Della Valle, E. 2008. RSWA 2008. Realizing a Semantic Web Application. Tutorial presented at the 7th *International Semantic Web Conference* (ISWC 2008), Karlsruhe - Germany.
Diederich, J., M. Dzbor, Maynard, D., 2007. REASE—The repository for learning units about the Semantic Web. *New Review of Hypermedia and Multimedia* 13(2): pp. 211-237.
Diederich, J., Nejdl, W., Tolksdorf, R., 2006. EASE: The European Association for SemanticWeb Education. In *Proceedings of the Semantic Web Education and Training Workshop (SWET'06), co-located with the First Asian Semantic Web Conference (ASWC 2006)*. China: Beijing.
Gerber, D., Frommhold, M., Martin, M., Tramp, S., Auer, S., 2010. Learning semantic web technologies with the web-based SPARQLTrainer. In *Proceedings of the 6th International Conference on Semantic Systems* pp. 26. ACM.
Giordano, D., 2004. Shared values as anchors of a learning community: A case study in information systems design. *Journal of Educational Media*, 29, 3, 213-227.
Hammar, K., and Sandkuhl, K., 2010. The State of Ontology Pattern Research - A Systematic Review of ISWC, ESWC and ASWC 2005–2009. In: Blomqvist, E., Chaudhri, V., Corcho, O., Presutti, V., Sandkuhl, K. (eds.) Proceedings of the *2nd International Workshop on Ontology Patterns* - WOP2010. CEUR-WS.org, Shanghai, China (2010).
Heath, T., Bizer, C., 2011. *Linked data: evolving the web into a global data space*. Morgan & Claypool Synthesis Lectures.
Horridge, M., 2009. A practical guide to building OWL Ontologies Using Protegé 4 and CO-ODE tools Edition 1.3 available at: http:// owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/ retrieved September 2013.
Huynh, D. F., Karger, D. R., Miller, R. C., 2007. Exhibit: Lightweight Structured Data Publishing..
Liyang, Y. 2011. *A Developer's Guide to the Semantic Web*. Springer.
McBride, B., 2002. Jena: A Semantic Web Toolkit. *IEEE Internet Computing* 6(6): pp. 55-58.
Neuhaus, F., Florescu, E., Galton, A., Gruninger, M., Guariono, N., Obrst, L., Sanchez, A., Vizedom, A., Yim, P., Smith, B., 2011. Creating the ontologist of the future. *Applied Ontologist*, 6, pp. 91-98.
Noy, N., Mc Guinness, D. L., 2004. Ontology development 101: A guide to creating your first ontology. Stanford University.
Poveda-Villalón M., Suárez-Figueroa M.C., and Gómez-Pérez A. 2010. A double classification of common pitfalls in ontologies. *Workshop on Ontology Quality (OntoQual 2010)*, Co-located with EKAW 2010, October 15, 2010, Lisbon, Portugal.
Rewctor, A., Drummond, N., Horridge, M., Rogers, J., 2004. OWL pizzas: practical experience of teaching OWL-DL: common errors & common patterns. Knowledge in the age, Springer.
Schmidt, M., Hornung, T., Lausen, G., Pinkel, C., 2009. SP2Bench: A SPARQL Performance Benchmark.
Schober, D., Grewe, N., Rohl, J., and Boeker, M. 2012. ZooAnimals.owl: a didactically sound example-ontology for teaching description logics in OWL 2. *OBML 2012 Workshop Proceedings*.