Integrating Semantic Web Features into Open-Source CMS Solution

Igor Vrdoljak, Nikola Bogunović
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10 000 Zagreb, Croatia
igor@netgen.hr, nikolabo@zemris.fer.hr

Abstract. *From e-commerce solutions to academia intranets and governmental online data sets, there is a growing need for introducing semantics into Internet and intranet sites. With main standards and building technologies defined and maintained by World Wide Web Consortium, the prospect of building "Web of data" seems good. Currently, one of main limiting parameters is the amount of such content available online, due to complex processes of publishing semantically enriched data. Most of semantic data available today comes from large governmental or industrial sources, with rather high barrier of entrance for smaller organizations often using general purpose content management tools. In this article we propose ways of extending widely used open source content management system (CMS) with a semantic layer in order to enable web editors to create Semantic Web data with minimal additional effort and training. An example implementation is proposed by utilizing open source CMS eZ Publish.*

Keywords. Semantic Web, CMS, Ontology, RDF

1 Introduction

Semantic Web or Web of Data represents vision of the next step in Internet evolution. In contrast to today's web of documents, which is mostly orientated towards human consumption, in such a scenario software agents would be able to understand and use information available on the Web. Machine readable content could be connected creating web of linked data representing different kind of concepts, from personal information to machine parts, bibliographical data and e-commerce related data. This data could be queried by using defined standard query languages, similar to databases being accessed through SQL as a common and standardized language. Finally, inference procedures could be utilized to reason over existing data, in order to find new and possibly interesting relationships between different concepts.

Currently, most of the development and information being introduced to the Semantic Web comes from large organizations (government, enterprises or online communities), that have acquired vast quantity of data and are starting to open it up [1]. Examples of these include large governmental projects like data.gov.uk and data.gov by governments of Great Britain and USA, aimed at presenting the public with data collected and produced by the government. Other is dbpedia.org, project started by by research groups from Universität Leipzig, Freie Universität Berlin, and OpenLink Software and aimed to extracting structured information from Wikipedia online encyclopedia and making it available for consumption on the web. On top of these resources, applications are created that combine information that is made available, analyze and present it, providing additional value and insight to the consumer.

There is still a large unexploited potential in enabling the smaller organizations and individuals typically using one of general purpose CMS solutions, to engage in creating and publishing semantic data. An approach which is mainly considered in addressing this problem is using specialized content management software with semantic features integrated in its' core. Examples of such systems include a number of Semantic Wikis, many of them requiring its end users to have expertise in Semantic Web technologies [2]. Also, as this approach requires for organizations to use

specialized content management tools, there is a probability that this would hinder the procedures of web content publishing and maintenance that are already in place. This presents a problem, especially for businesses, and could possibly cause the organization give up or slow down the integration of its data to Semantic Web.

In this article we present a different approach to the problem, in which a standard, general purpose content management system is extended and integrated into Semantic Web. In this way, organizations can keep using the software they are already accustomed to, thus keeping the training and adjustment costs low. An integration plan is proposed for an open source CMS - eZ Publish.

2 Related work

When addressing the problem of introducing the data to Semantic Web, a number of approaches emerged.

Some authors [3] are proposing introducing specialized content management solutions running RDF triplestores as back end, which in our opinion would introduce additional costs to the organizations already running traditional CMS software. Our aim is to lower the entry barrier, so we choose to utilize software already in use.

An approach more similar to ours is to introduce mapping between schema of relational databases that are at the heart of most of today's systems, and RDF / RDFs [4]. Triplify [5] is an example of such software, and offers a generic mapping tool for exposing relational data to the Semantic Web. This is an oversimplified approach for complex CMS solutions that store data in different back end systems and have additional content related features like access control policies.

The closest work comes from Drupal community [6]. Drupal, as of version 7 (at this time still under development), will offer out-of the box RDF output of some of the basic content stored in the CMS (like node titles, creation dates, comments count etc.). Set of specialized contributed modules is provided that can be used for defining and outputting additional semantic data from the structured content inside Drupal content store. SPARQL endpoint, which is also provided as a contributed module in Drupal 7, will enable external clients to access this semantic data in a standardized manner. For accessing external semantic data RDF Proxy module will be developed that will enable integration of external semantic web sources into Drupal installation. By this manner, information can be distributed and kept accurate across multiple sources on the Internet.

What is different in our approach is that we use existing eZ Publish content by binding it to RDF triples stored in the specialized triplestore server. This approach enables us to provide support to older installations already in production, avoiding the heavy manipulations on the content already available in the CMS. Also, due to intensive usage of specialized software to store and query RDF data, we hope to provide a more scalable solution, capable of working with large amounts of semantic data.

3 Semantic stack

Semantic Web, as envisioned by World Wide Web Consortium, consists of standards and tools that are organized in the Semantic Web Stack [7], shown in Fig. 1.

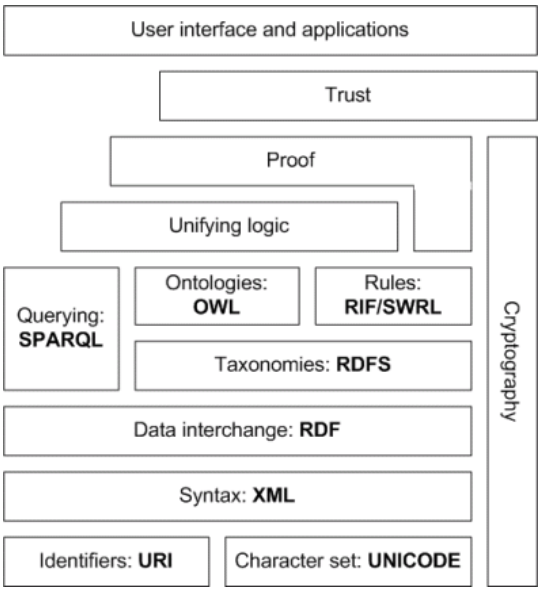

Figure 1: Semantic Web stack [7]

A layered nature of the technologies implies their hierarchical relation in which every layer exploits and utilizes capabilities of the layers below.

XML, as a well adopted language used thorough the Internet, enables users writing structured documents with a user-defined vocabulary encoded in XML Schema.

RDF (Resource Description framework) is a basic data model for writing simple statements about objects (resources) and relations among them. Although it does not rely on XML, and there are other serialization formats such as Turtle and N-triples, XML is the dominant format used on the Web.

RDF Schema (RDFS) [8] is the weakest ontology language available in the Semantic Web stack and provides modeling primitives such as classes and properties, subclasses and subproperties and domain and range restrictions. It is built upon RDF. Relation between RDF and RDF Schema is depicted in Fig.2.

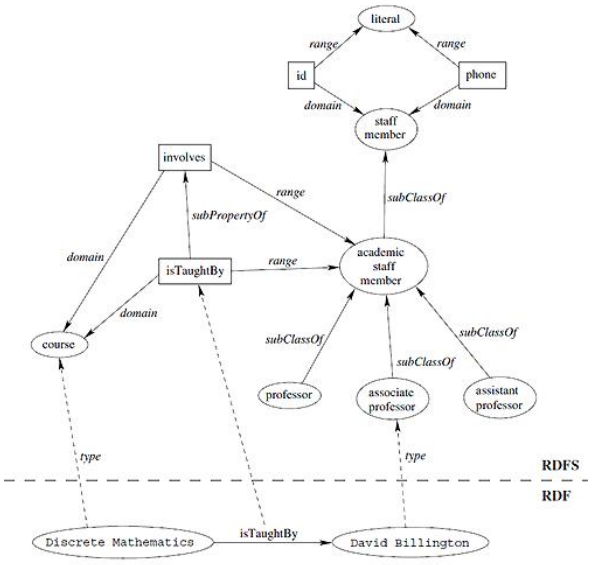

Figure 2: RDF and RDFS layers [7]

On the ontology layer OWL (Web Ontology Language) provides more power than RDFS in creating vocabularies by providing features like relation between classes (e.g. disjoiness), cardinality, equality, special characteristics of properties (like symmetry, transitivity or uniqueness), etc. Current version of OWL is OWL 2 [9], defined as W3C Recommendation. OWL 2 specifies three profiles (sublanguages) that are in fact trimmed down versions of OWL 2 that trade some expressive power for the efficiency of reasoning. Languages defined by these profiles achieve efficiency in a different way and are useful in different applications. OWL 2 EL is intended for use in applications that employ ontologies with very large numbers of properties and/or classes. OWL 2 QL is used in applications with large amounts of instance data, where query answering is the most important reasoning task. OWL 2 RL is aimed at applications that require scalable reasoning that can be implemented using rule-based reasoning engines, without sacrificing too much expressive power.

SPARQL is a RDF query language that can be used to query any RDF based data (including statements involving RDFS and OWL). Querying language is necessary to retrieve information for Semantic Web applications.

The top three layers of the Semantic Web Stack are still not standardized and are a subject of research.

In addition of putting semantic data on the web, interlinking to the already available datasets is needed in order to create Linked Data web in which pieces of information are connected through the web. To achieve this, set of expectations of behavior are in place, as suggested by Tim Burners Lee, one of the biggest proponents of Semantic Web [10]:

- Usage of URIs as names for resources on the web
- Usage of HTTP URIs so that people can look up those names.
- When someone looks up a URI, useful information is to be provided by using the standards available (RDF, SPARQL)
- Inclusion of links to other URIs is encouraged, in order to enable discovery of new and related concepts

4 CMS – working horse of today's web

In today's web of documents, most of the dynamic content available is published and

maintained through some kind of content management software. Content management system (CMS) market offers a broad spectrum of choices ranging from simple solutions dedicated for specific uses such as blogs and forums to enterprise level systems that provide advanced features like fine grained access control, integration to back-end systems and similar.

One of enabler features needed for outputting Semantic Web data from existing CMS installation is clean separation of data and presentation. Currently, many of the systems available still interweave content with design expressed in HTML and CSS markup making extraction and appropriate presentation of the data contained difficult and impractical. Some content management software in the market, like eZ Publish [13], enforce this separation very strictly, thus enabling content re-use and its representation in variety of different formats.

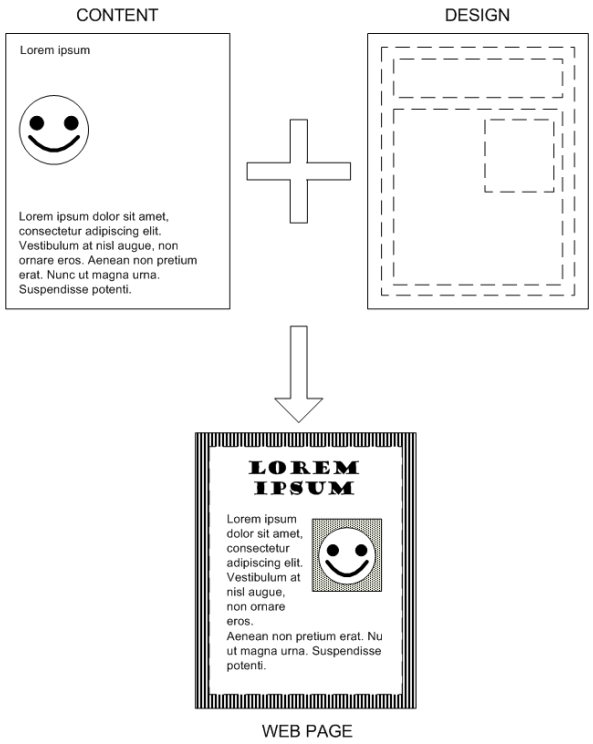

Figure 3: Separation of content and design in CMS [13]

Clean separation of content and presentation (design) as depicted in Fig. 3 makes the extraction and semantic presentation of data items feasible.

eZ Publish is a general purpose open source CMS developed and maintained by a Norwegian vendor eZ Systems. Although not among most popular CMS solutions available [11], partly due its perceived complexity and steep learning curve, it has a strong adoption among large organizations and companies that can put its advanced features to use. Such include strong customization and integration capabilities, advanced user management, fine grained access policy management, extensibility support and other.

One of the core features of eZ Publish, and the one which makes it a viable platform for extending towards semantic capabilities is its content model. It is inspired by a simplified version of object oriented programming paradigm in which concrete information is stored inside content objects. A content object is itself an instance of content class which defines its data structure.

Content class is made up of class attributes whose characteristics are determined by the data type that is chosen for that specific attribute. By combining different data types, it is possible to represent complex data structures. Data types available include simple data types like text line, checkbox (boolean-like data type), integer, float and complex data types like XML block, which is used to represent text enriched by simple formatting and capable of embedding relations to other content objects. An overall architecture of eZ Publish content model is depicted in Fig. 4.

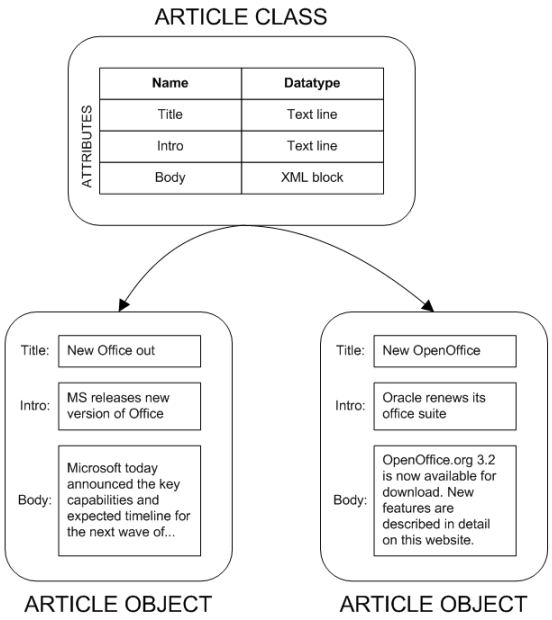

Figure 4: eZ Publish content object [13]

Relations between objects can be realized through specialized attributes that can represent single and multiple relations. In addition, relations between hierarchically organized concepts can be expressed by associating content objects with nodes in node tree, a mechanism which is used in CMS to hierarchically organize the objects that are present on the system.

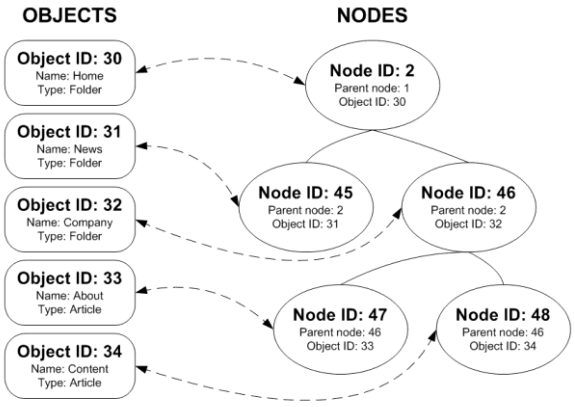

Figure 5: eZ Publish content objects and nodes [13]

Further, a well defined way of developing custom data types exists. Developers can define additional data types that integrate with those provided by the CMS out of the box.

All of these modeling features enable web site owners to define content models that represent knowledge on a certain domain in a structured manner. In following paragraphs we propose a way of exposing this knowledge for consumption on the Semantic Web.

5 Introducing Semantic Web features into eZ Publish CMS

5.1 Creating local ontology

When applying semantics onto existing or new installation of eZ Publish, we can distinguish between two levels of exposing data contained in CMS.

On the first level we can provide basic information regarding object properties and website structure which is inherently present in every installation. This information includes the basic meta-data available on content objects and their attributes, together with notion of relations existing between them. For example, the user that created the object can be exposed for every content object available on the system. Proposed default semantic data available for content objects is shown in Fig. 6.

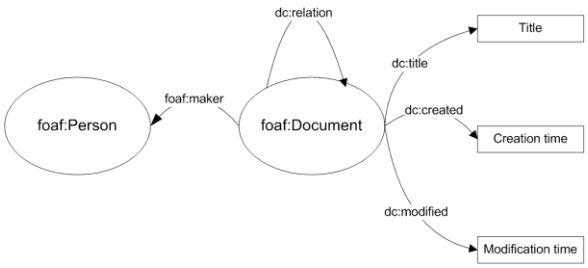

Figure 6: Default web page semantics expressed in RDF

For representing this basic information, we put in use existing ontologies such as FOAF [14] (Friend of a Friend) and Dublin Core [15].

FOAF is a well known ontology designed for describing persons, groups and organizations with their activities and relations to other objects. As it is one the most adopted vocabularies for such use, it is common sense to use it for describing different user profiles available in the CMS.

Dublin Core is another popular vocabulary that can be used to define metadata information for different document-like objects. The Dublin Core standard includes two levels: Simple and Qualified. Simple Dublin Core comprises basic fifteen metadata elements for describing resources (title, creator, subject, description, language, etc.). Qualified Dublin Core included three additional elements as well as a group of element refinements (qualifiers) that are used to refine the semantics of the elements.

We use foaf:Document and foaf:Person classes to represent basic information about the website page and the creator. Related objects (pages) that are defined by using eZ Publish relation attributes are represented by Dublin Core dc:relation property. Additionally, title of the page, as well as creation and modification dates are expressed with Dublin Core dc:title, dc:created and dc:modified properties. With this information we can have some exposition to Semantic Web and search engines that could interpret provided data, but most of the actual meaningful information stored in content objects still remains unavailable to machine consumption.

In order to present data to semantic agents in a more meaningful manner, we need a way of representing particular knowledge that is modeled in eZ Publish content classes.

On the second level we expose this specific data by using content model represented in content classes as a foundation for building up a specific ontology customized for the concrete web site use case. Ontology construction is in itself a complex problem which requires cooperation of different types of experts including both the domain and information technology experts. In our case we take the existing content model expressed with eZ Publish content classes as input for creating site ontology. What we need is a mapping of the content classes and their relationships into ontology entities that would be expressed in OWL or RDF Schema. This operation, performed by site administrator or developer needs to be done only once, and is valid as long as content model remains unchanged. After the definition of mapping rules, content generation performed by site editors remains unburdened with specifics of creating semantic content which is handled automatically by the system.

During creation of custom site ontology reuse of existing ontologies is highly recommended, especially those that are well defined and adopted by the online communities. This ensures better integration into Web of Data by providing meaningful information to wide range of existing agents.

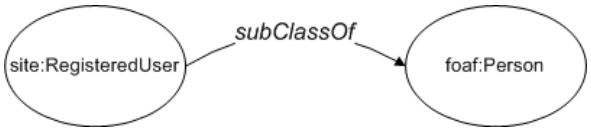

Figure 7: Subclassing foaf:Person class

Reuse is preferably done by importing external ontology and using *rdfs:subClassOf* and *rdfs:subPropertyOf* statements by which local ontology entities build upon external ones. This is done in order to ensure that specifics of local custom vocabulary do not propagate to external ontology. That being said, this does not detect or resolve all the inconsistencies that can happen due to the ontology reuse, created for example by conflicting cardinalities between the local and imported ontology. Due to this, a special attention is needed in the mapping process.

5.2 System architecture

To implement semantic features we propose a combined architecture that consists of eZ Publish CMS installation tightly integrated with specialized semantic framework such as Jena or RAP. This architecture provides us with best of the breed software for web content management and Semantic Web integration. Proposed system architecture is depicted in Fig. 8.

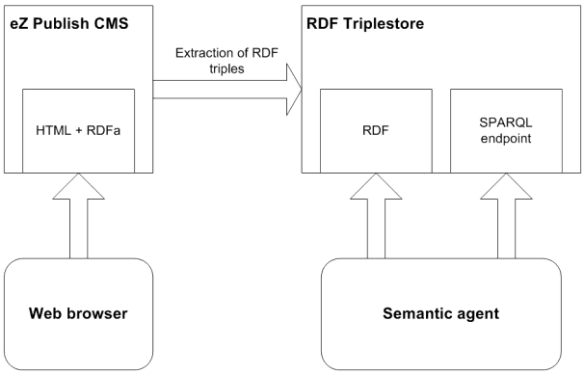

Figure 8: Overall solution architecture

In the proposed solution data is firstly entered into CMS by site administrators. Upon data creation, mapping function that relates CMS data model to specific site ontology is used to create RDF triples in the RDF triplestore. The same mapping function is also used to embed RDFa attributes into XHTML outputted by eZ Publish. Additionally, CMS data modifications and deletions are reflected in triplestore server. By utilizing trigger functionality of eZ Publish, which enables inserting custom workflows into publishing process, we can ensure consistency of CMS and triplestore data. Changes in security settings and access rights management also need to be taken in consideration to keep the data in the triplestore in sync with the CMS.

In this architecture different agents accessing the system receive appropriate output regarding on their capabilities and preferences. Web client expresses its preferred output by utilizing content negotiation mechanism already defined by the HTTP protocol [12] and gets redirected to the URL that serves the right type of content, as shown in Fig. 9.

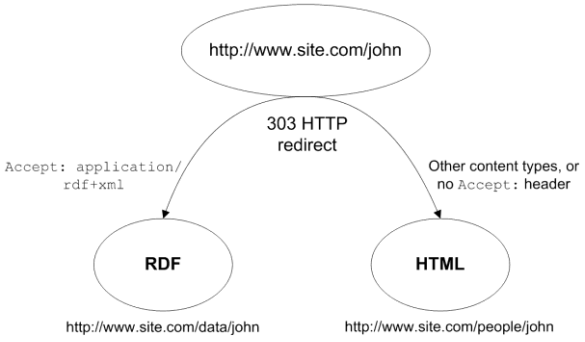

Figure 9: Redirection based on HTTP Accept: header

In addition to serving raw RDF data to the semantic agents, XHTML code is marked with RDFa attributes to provide semantics to clients that primarily use HTML such as web browsers.

RDFa (RDF in attributes) is a specification for using attributes to express structured data in any markup language. In case of XHTML [16] existing XHTML attributes, as well as a few of new ones are used to encode RDF statements into documents.

This data can be further used to augment the browsing experience by the web browser interpretation of such machine readable data and providing the user with additional information or capabilities. For example, theater performance info can be easily copied to the user's calendar if the browser correctly interprets date and location of the event expressed by XHTML attributes defined by RDFa and appropriate ontology.

For querying the system using SPARQL endpoint, a specialized RDF triplestore database, such as one provided by Jena framework, is used. This enables efficient querying of datasets containing large amount of RDF triples by avoiding putting the load on the CMS itself. Data extraction and importing mechanisms that use existing workflow functionalities in eZ Publish provide a way of maintaining the correctness of the data stored in the triplestore.

In addition of providing efficient access to RDF triples and a SPARQL query endpoint the usage of a specialized Semantic Web framework provides a base for building applications that can be used for reasoning over RDF triples stored in the triplestore. By utilizing inference engines integrated with toolkits like Jena or RAP additional facts can be inferred from RDF instance data and ontology information defined in RDFS or OWL. This continues to be a field for further research.

6 Conclusion

Recent actions taken by governmental agencies through the world, together with support of some of key online enterprises like Google, Yahoo and Facebook can serve as an indication that the idea of Semantic Web is finally gaining momentum. In this moment, one of the key enablers is to

provide simplified semantic data generation for a broad range of content producers using general purpose tools that need to develop semantic features.

In this paper we have proposed a way of enabling one enterprise level CMS solution to became first-class citizen in the yet to come Web of Data. By building on top of existing system we can keep the investments already put in place preserved and make the integration of already entered content as easy as possible.

Although we are considering specifically eZ Publish CMS in this paper, the recommended procedures could be applied to other content management systems with adequately advanced content model.

7 References

[1] SWEO Community Project -Linking Open Data on the Semantic Web : Statistics on Data sets Available at: http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics/, Accessed: 2nd May 2010.

[2] Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic Wikis, IEEE Software, July/August 2008, pp 8-11

[3] Staab S., Angele J., Decker S., Erdmann M., Hotho A., Maedche A., Schnurr H. P., Studer R.,Sure Y: Semantic community Web portals, Computer Networks, vol.33, issues 1-6, 2000, pp 473 – 491

[4] Stojanovic L. , Stojanovic N., Volz R.: Migrating data-intensive Web Sites into the Semantic Web, Proceedings of the 2002 ACM symposium on Applied computing, Madrid, Spain, March 11-14, 2002, pp.1100-1107

[5] Auer S., Dietzold S., Lehmann J., Hellmann S., Aumueller D.: Triplify - Lightweight Linked Data Publication from Relational Databases, WWW 2009 Madrid, 2009, pp 621 - 630.

[6] Corlosquet S., Delbru R., Clark T., Polleres A., Decker S.: Produce and Consume Linked Data with Drupal!, 8th International Semantic Web Conference (ISWC2009), 2009, pp 751 – 766

[7] Grigoris Antoniou, Frank van Harmelen: A Semantic Web Primer - second edition, The MIT Press, Cambridge, Massachusetts, USA, 2008.

[8] Brickley D., Guha R.V., McBride B: RDF Vocabulary Description Language 1.0: RDF Schema, available at: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, Accessed: 28th July 2010.

[9] W3C OWL Working Group: OWL 2 Web Ontology Language Document Overview, available at: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/, Accessed: 28th July 2010.

[10] Tim Berners-Lee: Linked Data, available at: http://www.w3.org/DesignIssues/LinkedData.html , Accessed: 2nd May 2010.

[11] Shreves R.: Open Source CMS Market Share, White paper, Water & Stone., available at: http://waterandstone.com/downloads/2008OpenSourceCMSMarketSurvey.pdf, Accessed: 2nd May 2010.

[12] Sauermann L., Cyganiak R., Völkel M.: Cool URIs for the Semantic Web, available at: http://www.dfki.uni-kl.de/~sauermann/2006/11/cooluris/, Accessed: 2nd May 2010.

[13] eZ Publish CMS documentation, available at: http://ez.no/doc, Accessed: 2nd May 2010.

[14] Brickley D.,Miller L.: FOAF Vocabulary Specification 0.97, available at: http://xmlns.com/foaf/spec/, Accessed: 28th July 2010.

[15] DCMI Usage Board: DCMI Metadata Terms, available at: http://dublincore.org/documents/dcmi-terms/, Accessed: 28th July 2010.

[16] Adida B., Birbeck M., McCarron S., Pemberton S.: RDFa in XHTML: Syntax and Processing, available at: http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/, Accessed: 28th July 2010.