


Use of Semantic Web technologies in Knowledge Management

Milos Jovanovik, Vladimir Apostolski, Dimitar Trajanov

Related papers

[Download a PDF Pack](#) of the best related papers 



[Europeana RDF store report](#)

Bernhard Haslhofer

[Semantic repository and ontology mapping](#)

Jochem Liem

[Knowledge Representation and Reasoning System \(KReS\)-Beta version report](#)

Valentina Presutti

USE OF SEMANTIC WEB TECHNOLOGIES IN KNOWLEDGE MANAGEMENT

Vladimir Apostolski
Faculty of Electrical
Engineering
and Information
Technologies – Skopje,
Rugjer Boskovic bb, PO Box
574, 1000 Skopje
Skopje, Macedonia

Riste Stojanov
Faculty of Electrical
Engineering
and Information
Technologies – Skopje,
Rugjer Boskovic bb, PO Box
574, 1000 Skopje
Skopje, Macedonia

Milos Jovanovik
Faculty of Electrical
Engineering
and Information
Technologies – Skopje,
Rugjer Boskovic bb, PO Box
574, 1000 Skopje
Skopje, Macedonia

Dimitar Trajanov
Faculty of Electrical
Engineering
and Information
Technologies – Skopje,
Rugjer Boskovic bb, PO Box
574, 1000 Skopje
Skopje, Macedonia

ABSTRACT

The fast technology development causes exponential increase of the number of information available to the people. But people do not to extract useful conclusions from all the information served, simply because they are either overloaded or uninterested about certain topics. By their nature, people are slow, unreliable, they forget and make errors while interpreting information. Machines, on the other hand, are much more effective in terms of speed, processing power and reliability, making them suitable for managing with human knowledge. The idea is to introduce systems for knowledge management, organization and interpretation by computers, which will raise the human-computer interaction level and improve distribution of knowledge according to individual needs. Systems capable of gathering, sharing, learning and interpreting stored information are referred to as Semantic Knowledge Management Systems. In this paper, we analyze traditional knowledge management software systems and then describe our effort to implement such knowledge management system that is fueled by available Semantic Web technologies. Its architecture is given in details, as well as the platform on which our experiments are performed. At the end of the paper we conclude that semantic knowledge management systems open a new field of opportunities for research and further realization of the idea about the Semantic Web.

I. INTRODUCTION

ONE OF THE BASIC human characteristics is to learn by gathering experiences. Those experiences can be earned individually, or by exchanging experiences with other people and media through which people communicate. With the wide range of possibilities for communication, the whole human knowledge increases rapidly. Due to the TV, printed media, mobile communications and the Internet, today's people are bombarded with information compared to the people who lived before these technological advances were discovered. Benefits of the global knowledge interconnection results with even faster technology development, because people need less time to reach solid level of knowledge and skills in a given domain. For example, today virtual communities exist, which can help in solving a variety of problems. The same solution can be reused numerous times by different people in different situations. Furthermore, big corporations, which own enormous amount of information, need efficient systems for managing that information. That is why knowledge

management systems were introduced in the first place. But as the number of information grows, people spend more time digging through search results in order to acquire the knowledge they need. The traditional text search is simply not efficient anymore, because every search result needs to be at least partially interpreted and its contextual meaning identified by the searcher. As a consequence, employees waste more of their working hours[1], reducing company quality of work and wasting financial goods. But this not only occurs in companies, but in everyday life as well, for example when one uses the search engines, like Google or Yahoo!. Because of the above mentioned phenomenon, a more efficient way to implement knowledge management systems is needed.

II. KNOWLEDGE MANAGEMENT SYSTEMS

A. Defining Knowledge Management

The term of knowledge management is inevitably bound to the terms of data, information and knowledge. Data is simply a fact. If the data is within given context, then it becomes information. Moreover, when implications can be predicted based on linking information together, one describes knowledge [2]. Different definitions for the term knowledge management exist. One of them is:

“Knowledge Management is the discipline of enabling individuals, teams and entire organizations to collectively and systematically capture, store, create, share and apply knowledge, to better achieve their objectives.”[3]

As Ron Young states [4], Knowledge Management (KM) involves 5 separated processes:

- Identify sources of information
- Create knowledge
- Store knowledge
- Share knowledge
- Use knowledge

B. Defining Knowledge Management Systems

According Young, the term Knowledge Management System (KMS) can be defined as a system capable planting common ground for the 5 KM processes to occur. Jafari and Akhavan define the term KMS as “ICT platform for collaboration and knowledge sharing with advanced services built on top that are contextualized, integrated on the basis of a shared ontology and personalized for participants networked in communities”[5]. Although not necessarily

related to computers, the term KMS is most often implemented with computational machines. Moreover, the Web 2.0 concept allowed these systems to further evolve around wiki software, forums, blogs and other collaboration software. Wikipedia is the most expanded knowledge management system developed. Although many argue the credibility of Wikipedia[6], it is still the most successful case study of collaboration software usage. Other examples are the individual or community blogs, where often news or how-tos are published. On the other side, as processes in companies become more complex, their employees spend more time in efforts of discovering and recalling of how some internal processes are performed. Therefore they also have some sort of internal KMS, but is not a rarity to publish some part of the KMS for the customer help as well. Often these corporative KMS include document processing, document search, collaborative editing, tagging, categorizing, chat and other communication features.

C. Knowledge Management Spectrum

KMSs, depending on the way knowledge is embedded into the system, can be classified in 6 categories[7][8]:

1. Transactional KM: Knowledge is embedded in technology.
2. Analytical KM: Knowledge is derived from external data sources, typically focusing on customer-related information.
3. Asset Management KM: Explicit management of knowledge assets (often created as a by-product of the business) which can be reused in different ways.
4. Process-based KM: The codification and improvement of business practice and the sharing of these improved processes within the organization.
5. Developmental KM: Building up the capabilities of the organization's knowledge workers through training and staff development.
6. Innovation/creation KM: Fostering an environment which promotes the creation of new knowledge, for example through R & D(research and development) and through forming teams of people from different disciplines

D. Knowledge Representation

Knowledge engineers, depending on the needs, should choose an appropriate representation of the knowledge, so that it will maximize the reusability, as well as minimizing the effort needed to capture the information, store it and share the knowledge by other community members. Knowledge representation, in more formal frames, is a part of Artificial Intelligence which should enable systems to perform logic and

reasoning.[9] In this context, traditional knowledge management systems do not literally have formal knowledge representation. The main challenge in the knowledge representation is enabling reasoning engines to further extend the information available by analyzing the existing facts from the knowledgebase, thus not requiring vast amount of time and memory. These requirements are often contradictory and finding such representation is far from trivial task.

Today's traditional KMS implementations involve usage of relational databases as persistent storage medium, which is very limiting in the aspect of expandability and interconnection of different information sources or KMSs. Relational databases have difficulties in expressing the means the metadata and the resources correlate. However, are widely used for techniques such as free text tagging or XML publications of metadata. Regarding the effort necessary to capture the information, a balance between the minimum effort needed in order to publish information and the usefulness of the information itself must be made[10]. Namely, the person who captures the information from its source, needs to put extra effort than he/she would normally do without storing and sharing it. It is a question of ethics of exactly how much effort should be put when gathering knowledge. Later in this paper, we will talk more about the formal representations of the knowledge, as ontologies[11], form which semantic knowledge management systems tend to use. However, according to Oren, Breslin and Decker[9], the sweet spot for balancing the effort and usability of information is between pure XML annotation and formal ontologies. We will cover Semantic Web based KMSs in more details later.

III. KMS PROBLEMS

Traditional approach in developing KMSs introduces various problems. First of all, efficiency and time needed for a given person to find certain information tends to increase, as topics overlap syntactically, leading to a situation where human inference is required in order to pick the most relevant information. Due to the rapid information count growth, the key terms start to appear in different contexts and meanings. This happens in both corporate KMS environments and even in ordinary web search engines. By far, computers are unable to determine the context in which terms might have higher relevance.

But the ever-increasing time required to come to the most relevant resource is not the only problem. Knowledge engineers experience heavy problems when trying to integrate two or more KMSs. Concepts in the knowledge representation of the KMS1 most often does not correspond with concepts representing the same objects in KMS2 objects. Therefore we conclude that KMS integration is very expensive task. Of course, web services may ease the integration as they are platform independent, but still complex mapping code logic must be written for integration process to be feasible. Furthermore, problems arise also when trying to expand knowing concepts with additional properties, mainly because of the low level of flexibility relational databases have. The only relation allowed in them, foreign keys, are very rigid and cannot explicitly express the type of relation between two

concepts. Table 1 Displays the differences between relational databases and ontologies.

Table 1. Comparison between relational database and ontology knowledgebase

Property	Relational Database	Knowledge Base
Structure	Schema	Ontologies
Data	Rows	Ontologies
Query Language	SQL	SPARQL
Relations	Foreign Key	Multidimensional
Uniqueness	Primary Key	URI

Because of the above described problems, it becomes apparent that new methods for knowledge representation are needed. In the following section, semantic knowledge management systems are introduced, as well as the benefits of their usage.

IV. SEMANTIC WEB AND KMS

The Semantic Web and the idea of Tim Berners Lee of building a web understandable by the machines has been very popular in the last years. Technologies and frameworks are constantly improving, by far essential components (such as Resource Description Framework – RDF, Web Ontology Language – OWL and query language SPARQL) are fairly stable. These advances have the potential to automate many processes that nowadays are performed either manually or human reasoning is needed. By assigning unique resource identifier – URI, machines are able to identify resources individually. This ability can bring large knowledgebases together, explaining thousands of concepts together with the relations between them, along with the flexibility needed for easy extension of the gathered knowledge. Semantic applications do not use only the information stored in their own storage, but they can rather pull information from other sources as well[12][13][14]. Due to the semi structured representation of the knowledge(ontologies), same concepts need not to be represented exactly with the same set of properties. The most powerful feature of the Semantic Web lies in building agents with capability to conclude new information based on the facts available in the knowledge bases the application is connected to. This process is also known as reasoning. Having the relations represented as triple statements(subject, predicate, object), querying semantic knowledge bases is fairly simple. By applying ontology alignment techniques, totally differently represented concepts can be considered as “the same” by the semantic reasoner engines, or strict rules can be defined, which reasoners can use to conclude new relations(statements) and add them in the knowledge base. When ontologies are used, information is stored in a form of graph, rather than in structured tables, such when using relational databases. Supported by the Linked Data [15] initiative, large information storages and vocabularies started to appear, such as DbPedia[16], Freebase[17], SIOC[18], MOAT[19] etc.

With these characteristics in mind, the Semantic Web technologies promise to overcome common issues mentioned in the traditional approach of knowledge management. For the first time in the history, computers are capable of accurately suggest and aid humans in inference, make task piping and predict the interests and relations among resources. The whole level of human-computer interaction is raised.

This technological advancement can also be used in knowledge management. By connecting to some of the large information sets mentioned, semantic applications for knowledge management can be developed. Increased personalization level of information is already being achieved with the Semantic Desktop movement[20], which suggests organizing information according to personal needs, and aligning it with external sources in order to pull finely grained information from them.

Moreover, these technologies enabled the popular wiki software to evolve into Semantic Wiki[21] software, where entities in articles are connected with relations, which in turn enables users to personally manage with their knowledge and pose queries to the internal search engine of the semantic wiki software that ordinary search engines are unable to answer. When using semantic wiki, each user can query the wiki engine for example, to retrieve all articles about books whose author is born after 1960. Obviously, the traditional wiki software is unable to give answer to such queries. The second shortcoming of the traditional wikis is that all the information is embedded with natural languages, rendering the reusability level of the information very low. [22] . Fig. 1 represents the dependency between effort and benefit when embedding metadata in wiki pages. It is clear that above certain level it becomes tedious for the user to embed more information, thus the benefit from it will saturate. Semantic Wikis’ potential for knowledge management has been brought to light, research groups are taken seriously and governments are finally ready to invest in semantic wiki development. The Federal Ministry of Education and Research of Germany got 2009 budget of 425,000€ [23] for the development of another wiki software, known as OntoWiki[24]. Unlike other semantic wiki systems, OntoWiki does not use special syntax for annotation, but rather tries to capture information by using forms instead. However, the most famous semantic wiki software is Metaweb’s Freebase. Freebase becomes increasingly attractive, to the level that Microsoft’s Bing claims it uses it when displaying query results.[25]. Namely, when user queries Bing for instance, “Pablo Picasso”, images of Picasso’s artwork might appear as well, or additional information, such as part of his biography.

Another large semantic knowledge base is DbPedia [http://www.dbpedia.org]. DbPedia extracts the information from Wikipedia articles and gives structured representation of that information. Moreover, DbPedia offers a SPARQL endpoint that allows users to search through the RDF graph DbPedia holds. It is interesting to note that Freebase and DbPedia become favored external knowledge bases when developing semantic web applications.[26]. However we made

efforts to implement our own semantic KMS. In the following section we describe the details.

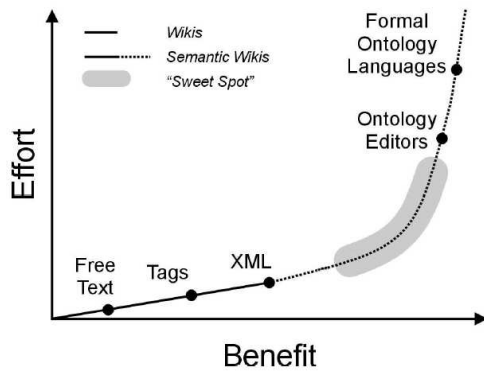


Fig. 1 Dependency between the effort and benefit when embedding metadata in wiki pages

V. IMPLEMENTING SEMANTIC KMS

Having in mind that suitable ground for developing Semantic Web application has emerged, we decided to deploy our own semantic KMS. First of all, we will cover the features that we aimed to fulfill, then we will discuss the Semantic Web framework we chose, and in the end we will talk about the demo application that we developed.

A. System features

The Semantic KMS we proposed, is supposed to act as semantically driven knowledge storage and sharing environment, with reasoning capabilities. Thus the system should be able to work with multiple information sources simultaneously, align the concepts within them, but also maintain information validity and disapprove any contradictory facts to enter the knowledge base. Finally, the KMS has to offer public web services for basic CRUD (Create, Retrieve, Update, Delete) operations.

B. Knowledge representation

The knowledge representation we chose is any form of OWL graph (file). Because of the flexibility of the Semantic Web development framework we chose, we could also possibly include RDF files here, but we mainly wanted OWL because of the expressiveness and reasoning capabilities it offers. The information is separated in two sections, schema and data. The schema contains the OWL class definitions and the relations between them. The data contains all the instances of the classes defined in the OWL schema.

C. Semantic Web Framework

With the evolving of the idea about the Semantic Web, frameworks for high level of abstraction began to emerge. These frameworks, mostly following the object-oriented paradigm, bring great reduction to the minimal effort needed to develop Semantic Web applications. Semantic Web frameworks usually consist of abstractions for three major components[21][27], storage, access and inference. Storage

components are usually RDF/OWL statements, whereas access components are query processors or application APIs that provide retrieval and information modification. The inference components are reasoning engines that apply interpretation of OWL semantics to the information in the knowledgebase. A knowledgebase is a collection of facts. The Semantic Web framework components serve to store, provide access to, and infer about these facts. Facts can be explicit or implicit. Explicit facts are those facts that have been directly added to the knowledgebase, while implicit facts are the facts whose existence is implied by the combination of explicit facts and semantics in ontologies and rules in the knowledgebase. Fig. 2 displays the modular design of the Semantic Web frameworks and the interconnection between their components. Basically all three components are connected between themselves. This means that often APIs offer way to query the knowledgebase and/or query the inference engine. On the other hand, the inference engine, after applying the inference rules, inserts entails new facts. Every insertion of a fact in the knowledgebase goes through checks for contradictions in the knowledgebase itself.

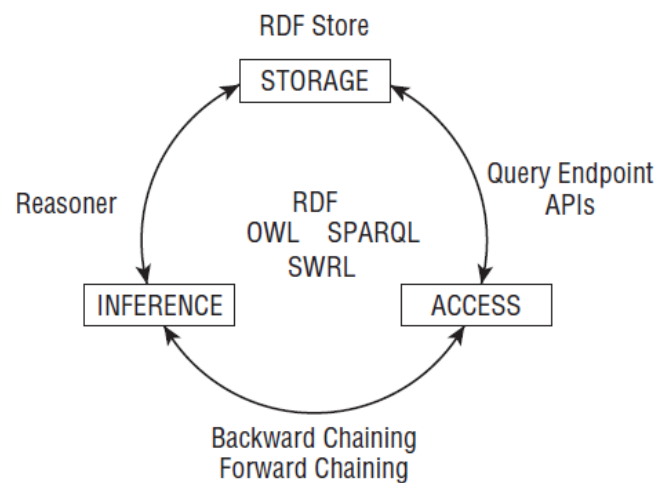


Fig. 2 Semantic Web frameworks consist of three components: storage, access and inference

One of the most stable and most widely used frameworks is the Jena framework[28], which we decided to use in the implementation of the Semantic KMS. Some authors propose using other frameworks, such as Sesame or Redland[21]. However, we decided to use the Jena framework, because it is one of the most stable and mature frameworks available in the time of writing of this paper. It is a Java framework, developed by Hewlett Packard (HP) Labs and it is an open source project. Basically, Jena provides Java environment for working with RDF, RDFS, OWL, SPARQL and reasoning engines. The Jena framework creates an additional layer of abstraction that translates the statements and constructs of the Semantic Web into Java artifacts, such as classes, objects, methods and attributes. One of the strongest sides of Jena lies in its excellent documentation. The exhaustive resources, including descriptions and tutorials that can be found on the Web encourage programmers to further develop their

Semantic Web applications utilizing this framework. As part of its RDF features, Jena offers managing with RDF resources, writing them in RDF/XML, N3 and N-Triples format. Jena also supports working with the RDF Schema, by providing API for all the vocabulary extensions it brings. Moreover, Jena covers the usage of OWL, in one of the three variants: Full, Description Logic, and Lite. Jena has a database backend API as well. It can be bound to SQL databases from many different vendors. Jena supports querying the model through the API, or by directly constructing SPARQL query to retrieve the results. The inference API includes several built-in reasoners, such as RDF(S), OWL, Transitive and Generic reasoners. Jena is also extensible with third-party reasoners, for example, with the Pellet reasoner. All reasoners support forward-chaining and backward-chaining modes, which can be configured individually.

D. System Architecture

In short, the systems consists of several modules: storage module, knowledge model module, reasoning module and end-point module. The storage module consists of RDF/OWL sources, either local or remote. These RDF/OWL sources allow pulling additional information to the local knowledgebase or importing portions of their graphs when retrieving triples. Once these triples are read, if relevant, can be inserted into the persistent storage backend. The decision whether some resources are relevant or not is brought by the knowledge model module. This module is compound of knowledge schema and loaded knowledge. Knowledge module essentially has two graphs. The first graph, knowledge schema, contains the OWL class and property definitions within the domain of interest. The second graph is the actual loaded knowledge, which can be queried or onto which inference engines can be run. The reasoning module contains the set of available reasoners which can be activated in order to extend the knowledge entailments within the knowledgebase. The end-point module is represented by set of web services that provide means for information retrieval and modification. Web services are chosen because we wanted the applications that will use the semantic knowledge base to be independent on the actual KMS implementation.

E. Platform Details

The proposed implementation of the KMS was performed on a Sony Vaio FW21E, with Core 2 Duo T8400 @ 2.6 GHz, 3 GB RAM, Windows 7 Ultimate operating system.

Regarding the software, we used Jena 2.6.0, NetBeans 6.7 IDE and Microsoft SQL Server 2008 relational database for the storage backend.

VI. PERFORMANCE RESULTS

The KMS we implemented contains 729 concepts in an OWL storage. The OWL source file size is 63KB. The ontology In the following sections we will present the results we got using different KMS (OWL) configurations and Jena resoners All reasoners are using forward-chaining mode. In average, it takes 0.5 seconds for the KMS to retrieve all triples, and 0.001 seconds to retrieve all triples where the subject is the resource AlienWare. There are 3 triples in the result set.

However, when inference engine is turned on, it takes 0.785 seconds to locate the AlienWare resource as subject in the knowledge base. In contrast, when inference engine is turned on, it takes 1.8 seconds to retrieve all 2686 triples in the inferred knowledge model. There are 15 triples in the result set, while the OWL mode is FULL.

However, if we switch the OWL mode to DL, the total number of triples is 2686. There are 17 triples in the result set. Approximate time to locate the resource AlienWare is 0.765 seconds. The approximate time needed to retrieve all triples from the knowledgebase is 1.748 seconds.

When switched to OWL Lite, the time needed to retrieve all 2686 triples from the knowledgebase is approximately 0.721 seconds.

When we switched the reasoner type from OWL to RDFS, it only took 0.003 seconds to retrieve the resource AlienWare, 8 results, 0.724 seconds to retrieve all 1396 triples from the knowledgebase.

By switching the reasoner to Transitive, the resource AlienWare was located in 0.001 seconds, only 3 results, 0.504 seconds to retrieve all 869 triples in the knowledgebase.

We will now try to analyze the gathered results. Please note that results may vary depending on the relations in the input ontology. According to the results we got, the OWL reasoner is most expressive, although it is the slowest reasoner. There are minor differences in measured time between different variants of OWL types(FULL,DL,LITE), although suggested literature claims the opposite [29]. However, these times could vary depending on the relations in the model. According to our results, the RDFS reasoner provides most suitable tradeoff between speed and expressiveness, keeping in mind that one sacrifices the benefits of the OWL-specific syntax.

A. Improving Retrieval Performances

Since the execution time of the retrieval of a specific resource might be unacceptable for some applications, we made efforts to improve the performance results, by applying a technique called graph closure[30]. Graph closure basically consists of calculating the new relations in the model by performing reasoning, and then comparing the initial model with the inferred model. Once they are computed, newly asserted facts are written to the knowledgebase. Next time the same resource is needed, one can simply query the model, without performing inference again, significantly decreasing the amount of time needed to retrieve the resource. In our case, the time needed to retrieve the AlienWare resource is 0.001 seconds when using the RDFS reasoner, and 0.002 seconds when using the OWL reasoner. This is a significant boost in performance when querying the knowledgebase. However, some authors propose other ways to improve the retrieval performances of the Jena framework, basically by partitioning the table in which it stores the triples in the database to several tables, according to their hierarchical classification[31]. It is not clear how to retrieve large datasets directly from Jena, since according to its documentation[32], it fetches the whole model in RAM at application startup, and then, when queried, iterates through it.

VII. FUTURE WORK

Although the results we got for our Semantic KMS might be encouraging, it is still unexplored how the system will behave in multiuser environment with large number of facts in the knowledgebase. Therefore we will continue to perform experiments using different ontologies both in size and number of relations between the concepts in the model, especially we will put effort in emphasizing the difference regarding performance between separate OWL dialects(FULL,DL,LITE). Also we will see how big the benefit will be from performing graph closures. In the future work, we will implement the activation of the reasoning engine as a process that will periodically occur and also implement caching techniques, to avoid multiple reboots of the knowledgebase for multiple web service calls. The KMS will be exposed to multiple applications simultaneously, in order to discover the system characteristics and prove the feasibility of the idea of this semantic KMS.

REFERENCES

- [1] Knowledge Management And Risk Strategies, Akira Ishikawa, Isamu Naka, Stallion Press, 2007
- [2] <http://www.systems-thinking.org/dikw/dikw.htm>
- [3] <http://knol.google.com/k/ron-young/the-future-of-knowledge-management/1emn5aby1s393/4#>
- [4] <http://knol.google.com/k/knowledge-management-back-to-basic-principles#>
- [5] A Review on Knowledge Management Discipline, Mostafa Jafari, Peyman Akhavan, Ashraf Mortezaei, Iran University of Science and Technology, Tehran 2009
- [6]http://www.createdebate.com/debate/show/Is_Wikipedia_a_credible_source[http://athena.libraries.claremont.edu/~blog/blog/archives/2005/12/15/wikipedia_-_a_credible_source.html]
- [7] <http://www.tlinc.com/articl51.htm>
- [8] Binney D., The knowledge management spectrum - understanding the KM landscape, Journal of Knowledge Management, 2001.
- [9]http://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning
- [10] How Semantics Make Better Wikis, Eyal Oren, John G. Breslin, Stefan Decker
- [11]http://en.wikipedia.org/wiki/Ontology_%28information_science%29
- [12] Approaches to Semantics in Knowledge Management, Cristiano Fugazza, Stefano David, Anna Montesanto, Cesare Rocchi, 2008
- [13] Semantic Web Technologies Trends and Research in Ontology-based Systems, John Davies, Rudi Studer, Paul Warren
- [14] Ontology and the Semantic Web, Robert M. Colombo, IOS Press, 2009
- [15] <http://www.linkeddata.org>
- [16] <http://www.dbpedia.org>
- [17] <http://www.freebase.org>
- [18] <http://www.sioc.org>
- [19] <http://www.moat.org>
- [20] Overview and Outlook on the Semantic Desktop
Leo Saueremann, Ansgar Bernardi, Andreas Denge
- [21] http://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki
- [22] Semantic Web Programming, John Hebel, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez, Wiley, 2009
- [23] <http://idw-online.de/pages/de/news300375>
- [24] <http://en.wikipedia.org/wiki/OntoWiki>
- [25] <http://www.freebase.com/solutions/bing>
- [26] Scripting Intelligence Web 3.0 Information Gathering and Processing, Mark Watson, Apress, 2009
- [27] Introduction to the Semantic Web and Semantic Web Services, Liyang Yiu, Chapman & Hall, 2008
- [28] www.jena.sourceforge.net
- [29] A Semantic Web Primer, Grigoris Antoniou and Frank van Harmelen, 2009
- [30] Introduction to Reasoning, Fulvio Corno, Laura Farinetti, 2007
- [31] An Efficient Web Ontology Storage Considering Hierarchical Knowledge for Jena-based Applications
Dongwon Jeong, Heeyoung Shin, Doo-Kwon Baik and Young-Sik Jeong, 2009
- [32] http://www.srdc.metu.edu.tr/webpage/documents/jena_doc/jena-faq.html#db-1