Creating and Managing Ontology Data on the Web: a Semantic Wiki Approach

Chao Wang, Jie Lu, Guangquan Zhang[1] and Xianyi Zeng[2]

[1] Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway, NSW 2007, Australia
{cwang, jielu, zhangg}@it.uts.edu.au
[2] Ecole Nationale Supérieure des Arts et Industries Textiles
9 rue de l'Ermitage 59100 Roubaix, France
xianyi.zeng@ensait.fr

Abstract. The creation of ontology data on web sites and proper management of them would help the growth of the semantic web. This paper proposes a semantic wiki approach to tackle this issue. Desirable functions that a semantic wiki approach should implement to offer a better solution to this issue are discussed. Along with that, some key problems such as usability, data reliability and data quality are identified and analyzed. Based on that, a system framework is presented to show how such functions are designed. These functions are further explained along with the description of our implemented prototype system. By addressing the identified key problems, our semantic wiki approach is expected to be able to create and manage web ontology data more effectively.

1 Introduction

Ontology has been realized to be an essential layer of the emerging semantic web [1, 2]. So the abundance of ontology related information is critical to the maturity of the semantic web. However, through the semantic web search engine Swoogle [3], while we've found that there are plenty of ontology schemas (which refer to classes, properties and their relations, as what "TBox" contains in description logics [4]) available over the web, ontology data (which refer to instances of classes or individuals as what "ABox" contains) do not seem to be very abundant in contrast.

One barrier for the problem is that it is not an easy job for ordinary users to create ontology data due to the complexity of the ontology languages, compared to the creation of normal web pages. Tools have been developed to assist the generation of ontologies with ease (e.g., Protege [5], OntoEdit [6], SWOOP [7]), but mostly they are not web-based tools and often used by ontology experts. In addition, while the ontology schemas, which usually reflect the domain knowledge, are relatively stable once developed, the ontology data are often prone to changes in a dynamic environment. Therefore, it may not be convenient to use offline tools to maintain such ontology data.

Recently, wikis [3] have been studied as an alternative way to create and maintain ontology data on the web. Several semantic wiki systems have been proposed. In contract to these studies, which mostly focus on some specific aspects or topics, this paper will first discusses the functions that a semantic wiki approach should implement to offer a better solution to web ontology data creation and management. Along with that, some key problems such as usability, data reliability and data quality will be identified and discussed. A system framework will be proposed to show how such functions are designed. These functions are further explained along with the description of our prototype system.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 discusses the desirable functions for managing ontology data and the framework design. Section 4 and Section 5 present in detail the fundamental functions of browsing, editing, search and query along with the description of our prototype system. Section 6 outlines the design and/or implementation of other functions. Section 7 concludes the paper and discusses the future work.

2 Related Work

This section first discusses the general ideas of wiki systems. Then related work on typical semantic wiki systems or frameworks is analyzed.

The main idea of most wiki systems is to let web users create and maintain web contents in a collaborative way. These systems provide convenient functions so that an ordinary user with little knowledge of web page making can contribute contents to the web. Wikipedia[4], the most well-known online collaborative encyclopedia on the Web, is a successful application of wiki systems. However, most contents maintained in wiki systems are semi-structured web pages only for web users to search and read. Facts and knowledge expressed in those web pages lack proper syntax and semantics for computers to process.

Semantic wiki systems then emerge to introduce semantic web standards (e.g., RDF/RDFS , OWL) and techniques to the common wiki systems, trying to make the collaborated contents have more semantics for computers to understand and process. We will discuss these systems as follows.

Platypus Wiki [8] is an extended wiki system that allows users to input RDF and OWL statements in addition to plain text. However, how the system helps user to create such semantic statements has not been discussed in detail in [8].

Powl [9] is a web based platform that provides a collaborative environment for semantic web development. OntoWiki [10], which is built upon Powl, employs web 2.0 techniques to bring more features in authoring, editing, and searching semantic contents. Unlike Powl and OntoWiki which concentrate more on ontology editing, Semantic Wikipedia [11], as an extension to the popular MediaWiki software, brings semantics to texts and links in its wiki system. These systems usually have better user interfaces than those discussed before.

[3] http://en.wikipedia.org/wiki/Wiki
[4] http://www.wikipedia.org

There exist more other semantic wiki systems (e.g., WikSAR [12], COW [13], WikiFactory [14] and etc), which provide more features such as integration with the desktop [12], typical query mechanisms [13], domain-oriented design [14], and etc.

3 Design Considerations

3.1 Desirable Functions for Managing Web Ontology Data

The previous section shows that the existing work on semantic wikis is mostly focused on some specific topics. In contrast, we first discuss some desirable functions that are commonly required for managing web ontology data. By doing so, we try to provide some hints towards the design and development of better and more extendable semantic wiki systems.

Browsing, creating and editing. These functions involve the provision of facilities and interfaces for browsing the ontology schema and data, creating and updating the data as required. These functions are actually the most fundamental functions for any types of semantic wikis. However, how such functions are implemented matters. Since the formats of ontology data (e.g., OWL) are not intuitive to ordinary users, it is desirable to design these functions with more intuitive and interactive features.

Search and Query. Obviously, these functions help to retrieve created ontology data from the web site in response to the information need of the users. Generally, the search function may refer to some form of query upon free/unstructured texts such as the search provided by web search engines. In contrast, the query function is usually associated with the structured data (i.e., SQL for the relational databases). Since web ontology data is kind of semi-structured data, it is desirable to have both of these functions in semantic wikis.

Authentication and access control. These functions deal with the identification of the user and the control of their operations on the web ontology data. Existing semantic wiki systems seem to lack a sophisticated mechanism to implement this type of functions. The lack of this mechanism affects the trust on the contents from people, which hinders their use in some serious situations.

Data quality control. Functions of this type ensure the overall web ontology data contributed by users are of accepted quality. One issue of quality is that the data are not reliable. The deployment of proper functions on "authentication and access control" could help handle this issue as malicious editing or spam could be screened off by it. Another common issue is that the duplicated data may be generated in a distributed Web environment, which also decreases the overall data quality. In addition, incompleteness and inaccuracy are threats to data quality as well.

3.2 The System Framework

This subsection overviews the framework based on which the prototype system "robinet" is developed. It shows the general structure of the system and the relations among the functions. The system is implemented in Java.

Fig. 1 illustrates the general structure of the robinet system and the interaction between it and its managed web ontology data and users. At its lowest layer, the data layer, access rules and indexed data are stored in addition to the managed ontology data. Above the data layer is the API layer, which is made up of low level packages and the APIs they offer. Based on these APIs, a supporting layer provides customized tools which are used to build various operations, data models and functions for presentation. The propose of this layered design is to make the implemented system easy to be maintained and extended.

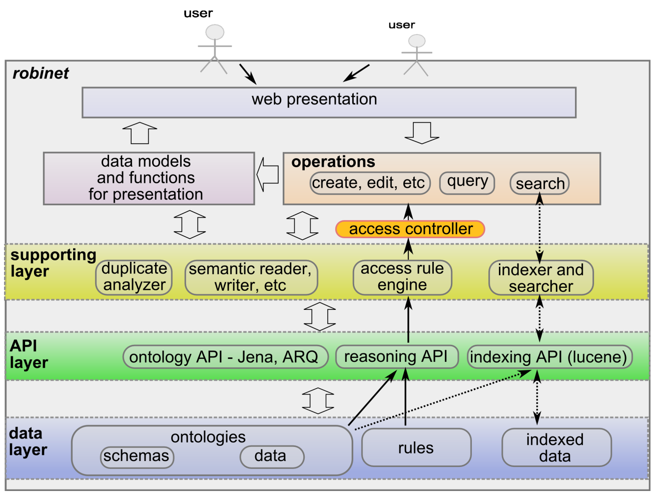


Fig. 1. The framework for the prototype system “robinet”

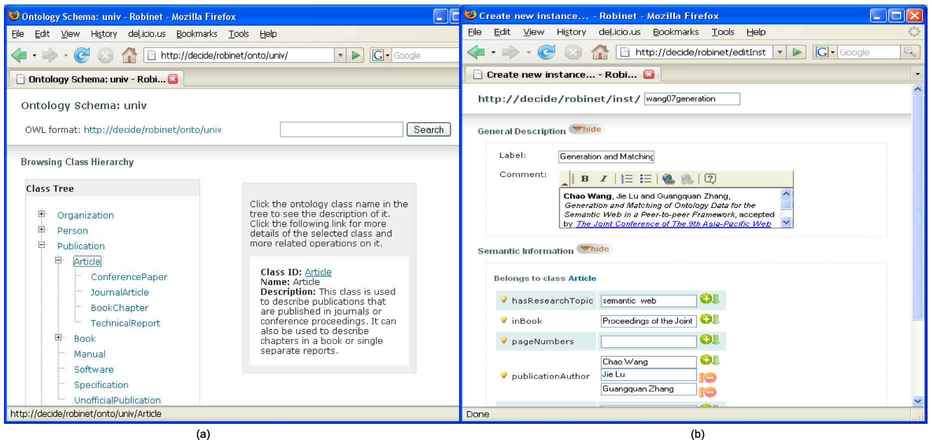
The framework is designed with principles such as modularity, friendly URLs, and ease of use. Along with these design principles, we have implemented essential functions to help web ontology data creation and management. Such functions allow users to browse ontology schemas, create and edit instances and etc. The following sections will discuss them in detail.

4 Browsing, Creating and Editing

4.1 Ontology Schema Browsing

Browsing ontology schemas helps ordinary users get familiar with the domain that the ontology schemas are intended to describe. It is clear that a user without an adequate understanding of the domain ontology schema may not be able to proceed and create the correct information for the domain. Some ontology specific editing tools (e.g., Protege [5], and etc) can provide graphical user interfaces that allow users to view the domain ontology schemas intuitively. But this requires users to install additional software. In addition, since such editing tools are mostly designed for domain experts or certain professionals to develop ontologies in a desktop environment, they are not web-based solutions preferred by ordinary users. Therefore, it is desirable to have a web-based browsing mechanism so that users can just use any types of browsers to learn the domain ontology schemas before they are ready to contribute information.

Initially, the "robinet" system requires a setup process such as loading the ontology schemas for browsing. Once the initial setup is done, users can view the ontology schemas using common browsers. Fig. 2 (a) shows the web interface which renders the classes (concepts) in an ontology schema about the academic domain in a tree structure according to their is-a relationship. Users can get familiar with these classes and their relations by exploring this tree. In addition, if they are interested in a particular class, they can click it in the tree to see detailed comments (if supplied by the schema makers) on this class in the right-hand side panel. If further information is needed to know about this class, users can follow the link in the right-hand side panel to view such information.



Ontology Schema: univ - Robinet - Mozilla Firefox
File Edit View History del.icio.us Bookmarks Tools Help
http://decide/robinet/onto/univ/
Google
Ontology Schema: univ - Robi...
Search
Class Tree
http://decide/robinet/onto/univ/Article
(a)
Create new instance... - Robinet - Mozilla Firefox
File Edit View History del.icio.us Bookmarks Tools Help
http://decide/robinet/editInst
Google
Create new instance... - Robi...
hide
hide
Done
(b)

Ontology Schema: univ

OWL format: http://decide/robinet/onto/univ

Browsing Class Hierarchy

- Organization
- Person
- Publication
  - Article
    - ConferencePaper
    - JournalArticle
    - BookChapter
    - TechnicalReport
  - Book
  - Manual
  - Software
  - Specification
  - UnofficialPublication

Click the ontology class name in the tree to see the description of it. Click the following link for more details of the selected class and more related operations on it.

Class ID: Article
Name: Article
Description: This class is used to describe publications that are published in journals or conference proceedings. It can also be used to describe chapters in a book or single separate reports.

http://decide/robinet/inst/ wang07generation

General Description

Label: Generation and Matching

Comment: **Chao Wang**, Jie Lu and Guangquan Zhang, *Generation and Matching of Ontology Data for the Semantic Web in a Peer-to-peer Framework*, accepted by *The Joint Conference of The 9th Asia-Pacific Web*

Semantic Information

Belongs to class Article

hasResearchTopic	semantic web
inBook	Proceedings of the Joint
pageNumbers	
publicationAuthor	Chao Wang
	Jie Lu
	Guangquan Zhang

Fig. 2. The interfaces for browsing domain ontology schema and editing ontology data

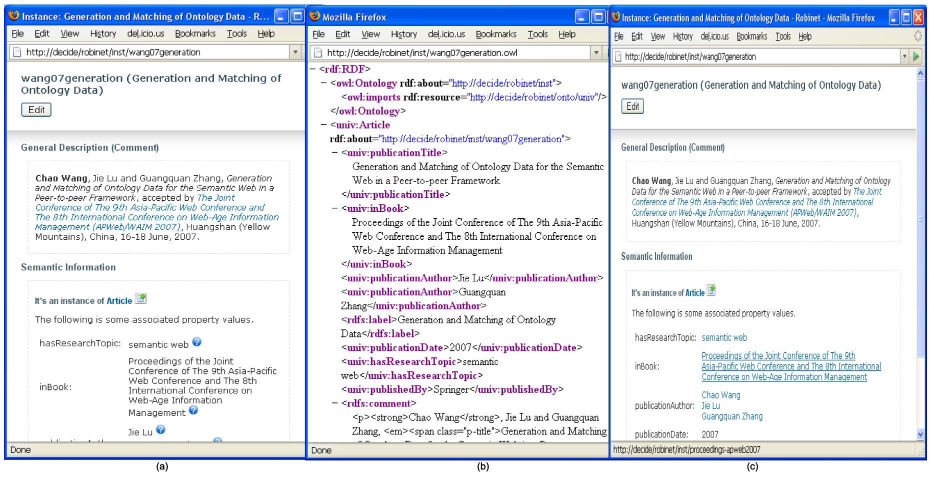
In the system, the URLs used to identify their corresponding resources (classes or properties) is quite friendly to both end users and other semantic web applications which may run at other computers in the network. For example, while the URLs http://decide/robinet/onto/univ/ and http://decide/robinet/onto/univ/Article lead to human-readable web pages about the ontology schema `univ` and the its class `Article` respectively, the URLs http://decide/robinet/onto/univ and http://decide/robinet/onto/univ#Article identify the original OWL formatted versions which are more machine-understandable. Therefore, other semantic web applications can use such URLs to import the interested ontology schemas into their application space remotely. This mechanism could help the development of semantic web applications and intelligent software agents over the web.

4.2 Ontology Data Creation and Editing

Once users have become familiar with the ontology schemas by browsing them, they can create new ontology data through the functions of the system. Currently, two main types of functions are implemented. We will discuss them respectively.

The first type of functions enables user to create completely new ontology data. For example, a staff, if he/she wants to publish some data about his/her recent publications, can choose the class that is appropriate for the data. He/She may choose the class `Article` in the ontology schema `univ` and use it as the type for the new instance created to annotate a paper. Fig. 2 (b) demonstrates the interface used for this operation. An unique ID is required from the user to form the URL used to identify this publication.

Fig. 3 (a) shows the created instance in a human-readable view, which can be accessed through the URL generated based on the user-supplied ID. Actually, this view is generated by the system from the OWL formatted ontology data which can also be accessed through a browser, as shown in Fig. 3 (b). Obviously, the actual ontology data is not comfortable for ordinary users to read. However, computers on the other hand is good at processing such data. Since they are aligned with the given ontology schema, well designed semantic web applications that are aware of this domain ontology can use them to perform further complicated tasks.


Instance: Generation and Matching of Ontology Data - R...
File Edit View History del.icio.us Bookmarks Tools Help
Mozilla Firefox
Instance: Generation and Matching of Ontology Data - Robinet - Mozilla Firefox
Edit
Done
(a)
(b)
(c)

http://decide/robinet/inst/wang07generation

wang07generation (Generation and Matching of Ontology Data)

General Description (Comment)

Chao Wang, Jie Lu and Guangquan Zhang, *Generation and Matching of Ontology Data for the Semantic Web in a Peer-to-peer Framework*, accepted by *The Joint Conference of The 9th Asia-Pacific Web Conference and The 8th International Conference on Web-Age Information Management (APWeb/WAIM 2007)*, Huangshan (Yellow Mountains), China, 16-18 June, 2007.

Semantic Information

It's an instance of Article

The following is some associated property values.

hasResearchTopic:	semantic web
inBook:	Proceedings of the Joint Conference of The 9th Asia-Pacific Web Conference and The 8th International Conference on Web-Age Information Management
	Jie Lu

http://decide/robinet/inst/wang07generation.owl

```
– <rdf:RDF>
  – <owl:Ontology rdf:about="http://decide/robinet/inst">
      <owl:imports rdf:resource="http://decide/robinet/onto/univ"/>
    </owl:Ontology>
  – <univ:Article
    rdf:about="http://decide/robinet/inst/wang07generation">
    – <univ:publicationTitle>
        Generation and Matching of Ontology Data for the Semantic
        Web in a Peer-to-peer Framework
      </univ:publicationTitle>
    – <univ:inBook>
        Proceedings of the Joint Conference of The 9th Asia-Pacific
        Web Conference and The 8th International Conference on
        Web-Age Information Management
      </univ:inBook>
      <univ:publicationAuthor>Jie Lu</univ:publicationAuthor>
      <univ:publicationAuthor>Guangquan
      Zhang</univ:publicationAuthor>
      <rdfs:label>Generation and Matching of Ontology
      Data</rdfs:label>
      <univ:publicationDate>2007</univ:publicationDate>
      <univ:hasResearchTopic>semantic
      web</univ:hasResearchTopic>
      <univ:publishedBy>Springer</univ:publishedBy>
    – <rdfs:comment>
        <p><strong>Chao Wang</strong>, Jie Lu and Guangquan
        Zhang, <em><span class="p-title">Generation and Matching
```

http://decide/robinet/inst/wang07generation

wang07generation (Generation and Matching of Ontology Data)

General Description (Comment)

Chao Wang, Jie Lu and Guangquan Zhang, *Generation and Matching of Ontology Data for the Semantic Web in a Peer-to-peer Framework*, accepted by *The Joint Conference of The 9th Asia-Pacific Web Conference and The 8th International Conference on Web-Age Information Management (APWeb/WAIM 2007)*, Huangshan (Yellow Mountains), China, 16-18 June, 2007.

Semantic Information

It's an instance of Article

The following is some associated property values.

hasResearchTopic:	semantic web
inBook:	Proceedings of the Joint Conference of The 9th Asia-Pacific Web Conference and The 8th International Conference on Web-Age Information Management
publicationAuthor:	Chao Wang Jie Lu Guangquan Zhang
publicationDate:	2007

http://decide/robinet/inst/proceedings-apweb2007

Fig. 3. The created instance of the class `Article`. (a) The instance shown in a human readable view; (b) The actual ontology data created by the user. (c) The instance after related instances are created and automatically linked with it.

After instances are created, they can be further edited by clicking the "Edit" button as shown in Fig. 3 (a).

The second type of functions allows users to create new ontology data which are related to the existing data.

First we discuss why we need this type of functions. This type of functions rely on the existence of a type of properties called "object properties" which link one instance to another with a particular semantic relation [15]. They offer the functionality similar to hyperlinks which are used over the current web to link different web pages together. The importance of hyperlinks has ready been obvious. Without them, The web could not form and grow into such a huge connected information repository as currently is; nor could modern search engines build a large information base and rank the related information with certain measurements such as authority or popularity [16][17]. Similarly, it is also very important to use object properties to link instances together. The resulting relational feature in ontology data could be even more essential than using normal hyperlinks in web pages as it delivers more specific semantic meanings than normal hyperlink mechanism. Therefore, it is very desirable to enable users to create ontology data which are linked to other ontology data via certain object properties.

Then we show how easy it is to create related new ontology data based on existing ontology data in our system. As explained before, when a completely new instance is created, initially all the property values are supplied as strings by users, regardless of the types of properties. However, after the instance is created, the system will detect according to the ontology schema whether a property value should be a certain data type or an instance. For those who should be instances via given object properties, it provide the function to create them. As shown in Fig. 3 (a), those property values that should be instances have question marks associated. By clicking the question mark, users will be presented an instance creation interface which is similar to the one shown in Fig. 2 (b). After creating these new instances, they are related to the existing instance with the specified object properties. Fig. 3 (c) shows the web pages of the existing instance after its related instances are created. The question marks has disappeared. Instead, those object property values are linked to the web pages of it's related instances.

5 Search and Query

As mentioned earlier, we distinguish search and query as two types of retrieval functions with their different focuses. The following discusses them respectively.

5.1 Free Text Search

Free text search doesn't require users to know much about query languages. One or more keywords are enough as input to invoke a search, just like the experience of using a web search engine. This seemingly basic function has become pervasive. For a conventional web site especially a web portal, having this type of search function offers users a simple and straightforward way to find and locate information. The same is applied to a semantic web site (or portal) that hosts ontology data.

Currently, we implement the free text search function using Lucene [5], a convenient java package for indexing and searching. In our implementation, every instance is treated as a virtual document. Its properties (including annotation information such as labels and comments) are concatenated together to form the contents of the document. This virtual document is then tokenized and indexed, therefore, able to be searched.

5.2 Structured Query

Structured queries are very useful to retrieve specific information out of the data source. This usually requires the knowledge of the data structure or the schema. Users are also required to know certain query languages that are used to issue queries. SPARQL [18] is one of the languages designed for the query of semi-structured RDF data like ontology data. We use ARQ [6], a SPARQL engine to process queries upon the created ontology data. Users are assumed to know about the query language. By browsing the ontology schema through the system, they could get familiar enough with it to compose their queries to get desired data.

Since most users are reluctant to learn specific query languages, it would be better to use structured query in some indirect ways other than this straight way. Similar to the idea in [13], queries can be specified by some competent users and they can be embedded in certain web pages. This mechanism allows the creation of typical and useful views on the ontology data. Furthermore, external web related applications or web agents can use the query mechanism to exploit the repository of the web ontology data effectively.

6 Discussion on Other Functions

Due to limited space, this section briefly outlines the design and/or implementation of the rest of the two functions as follows.

Authentication and Access Control. Some serious situations require authentication and access control for web ontology data management. We exploit the semantics of the ontology and use a set of access rules to provide a flexible mechanism for the implementation of this type of functions. To do so, we treat the system as a model with tuple $\Lambda(C, O, P, R)$ where C denotes *Semantic Contents*, O denotes *Operations*, P denotes *Participants* and R denotes *Access Rules*. In the system, C, O and P are all modeled using ontology. R is then composed using the elements from these three sets. When an access to the system happens, the access pattern is identified and reasoned against the rules together with the ontology to determine whether the access is accepted or rejected. Fig. 1

[5] http://lucene.apache.org/
[6] http://jena.sourceforge.net/ARQ/

also shows the general components that are designed to work together for this type of functions.

Data Quality Control. One major issue related to data quality is data duplication and overlapping, which often happens when many users contribute their data simultaneously. To tackle this issue, first the duplicated data should be detected. We have developed particular methods [19][20] to accomplish this task. our methods explore the features of ontology data, therefore, achieve better results compared to other conventional methods according to our experiments. We are currently in the process of integrating these methods into the system. So the system will be able to detect potential duplicated instances created by different users. They will be labeled (for example using the annotation property `rdfs:seeAlso`) for further review from users. Once the positive feedbacks are made by users, they can be further labeled with the tag `owl:sameAs`. When multiple duplicates are found for one instance, it is then possible to determine which instance is the most complete and reliable. Therefore, it also helps to solve the problem of data incompleteness.

7 Conclusions and Future Work

This paper proposed a semantic wiki approach for managing ontology data on the Web. It discussed the desirable functions a semantic wiki should have to manage the data. A framework were presented to illustrate the design of such a semantic wiki system. These functions were then further explained and demonstrated in regard to the implementation of this framework. In particular, fundamental functions of our implemented prototype system were discussed in detail to demonstrate the usability of the semantic wiki approach to web ontology data creation and management.

As an ongoing project, the future work includes the further improvement of the system interaction by using more advanced web techniques and the tighter and better integration with other function modules, for example, the function for data quality control. How to utilize the managed ontology data to provide certain services through web applications or agents will also be studied.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (2001) 34–43
2. Hendler, J.: Agents and the semantic web. Intelligent Systems, IEEE **16** (2001) 30–37
3. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management, ACM Press (2004) 652–659
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The description logic handbook : theory, implementation, and applications. Cambridge University Press, New York (2002)

5. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubezy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of protege: an environment for knowledge-based systems development. Int. J. Hum.-Comput. Stud. **58**(1) (2003) 89–123
6. Sure, Y., Erdmann, M., Angele, J., Staab, S., Wenke, R.S.D.: Ontoedit: Collaborative ontology development for the semantic web. In: Proceedings of the 1st International Semantic Web Conference, Sardinia, Italy. (2002)
7. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B., Hendler, J.: Swoop: A 'web' ontology editing browser. Journal of Web Semantics **4**(2) (June 2006) 144–153
8. Tazzoli, R., Castagna, P., Campanini, S.E.: Towards a semantic wiki wiki web. In: Proceedings of the 3rd International Semantic Web Conference. (2004)
9. Auer, S.: Powl - a web based platform for collaborative semantic web development. In: Proceedings of the First Workshop Scripting for the Semantic Web. (2005) http://www.semanticscripting.org/SFSW2005/papers/Auer-Powl.pdf.
10. Auer, S., Dietzold, S., Riechert, T.: Ontowiki - a tool for social, semantic collaboration. In: Proceedings of the 5th International Semantic Web Conference, Springer (2006) 736–749
11. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic wikipedia. In: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 585–594
12. David Aumueller, S.A.: Towards a semantic wiki experience c desktop integration and interactivity in wiksar. In: Proceedings of the 1st Workshop on the Semantic Desktop in conjuction with the 4th International Semantic Web Conference. (2005)
13. Fischer, J., Gantner, Z., Stritt, M., Rendle, S., Schmidt-Thieme, L.: Ideas and improvements for semantic wikis. In: Proceedings of the 3rd European Semantic Web Conference. (2006) 650–663
14. Iorio, A.D., Presutti, V., Vitali, F.: Wikifactory: a web ontology-based application for creating domain-oriented wikis. In: Proceedings of the 3rd European Semantic Web Conference. (2006)
15. McGuinness, D.L., Harmelen, F.v.: Owl web ontology language overview. w3c recommendation. http://www.w3.org/tr/2004/rec-owl-features-20040210 (2004)
16. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In: Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms. (1998)
17. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the seventh international conference on World Wide Web. (1998) 107–117
18. Prudhommeaux, E., Seaborne, A.: Sparql query language for rdf (http://www.w3.org/tr/rdf-sparql-query/) (2007)
19. Wang, C., Lu, J., Zhang, G.: Integration of ontology data through learning instance matching. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. (2006) 536–539
20. Wang, C., Lu, J., Zhang, G.: A constrained clustering approach to duplicate detection among relational data. In: Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2007) 308–319