

Ontology-based constructing and maintaining of Wiki-systems

V. K. Shestakov, Yu. A. Zagorulko

Abstract. The paper discusses the problems of the development and maintenance of information systems on the basis of ontology and Wiki-technology. In particular, an approach including the method of constructing information systems using Wiki-technology and ontology of subject domain and the method of ontology extraction from the already existing Wiki-systems is suggested. An implementation of these methods is described in detail.

Keywords: ontology, Wiki, semantic technology, information system construction, data and knowledge extraction.

1. Introduction

To satisfy the constantly growing information requirements of users, various tools for the creation of information systems are developed. Now tools based on Wiki-technology are very popular due to the many advantages of this technology. In particular, Wiki-technology enables us to operate with both text and multimedia content. It provides a handy and intuitive user interface, so the technology is easy to operate. Wiki-technology, however, has an important shortcoming: in developed information systems, it allows us to control only a structural integrity of references and does not provide a logical integrity or semantic consistency of the concepts used in such systems. The idea of the approach suggested is to create tools able to provide the development of information systems with a consistent concept system (the semantically consistent Wiki-systems). A Wiki-system with such features can be obtained if it is developed on the base of a logical consistent ontology describing a subject domain of a future system.

The paper presents an approach to the development of information systems on the basis of the subject domain ontology. With a view to using this approach for the maintenance, evolution and reengineering of the existing Wiki-systems, it is supplemented with the method of extracting ontologies from Wiki-systems. In addition, this method may be used to facilitate the development of a new Wiki-system on the basis of several Wiki-systems.

2. Related works

Today, there are a lot of projects devoted to building Wiki-systems using semantic technology. Although these projects are focused on solving different

tasks, all of them use ontology at the various stages of building or operating an information system.

The project OntolingWiki [1] offers tools for building corporative Wiki-portals. The structure of such portal is completely specified by the ontology which is used as its basis. The ontology provides the users with an additional way to navigate through the portal content. The main shortcoming of this project is that the ontology cannot be modified with the help of the facilities of the Wiki-system itself.

The most common tool used for the creation of Wiki-systems is Wiki-engine MediaWiki [2]. One of the most popular semantic extensions for it is Semantic MediaWiki [3, 4]. Initially, it was assumed that Semantic MediaWiki would enable us to import the ontology [5] that would specify the structure and relations of the Wiki-system to be developed. However, this function was presented only in its beta-version and is missing in current releases.

Another MediaWiki extension is BOWiki [6], which is a semantic Wiki-engine based on ontology. The advantage of BOWiki is that it allows users to treat entities defined by Wiki-pages as the instances of ontological categories, to define new relations in Wiki-content, to link Wiki-pages by n -ary semantic relations, and to request Wiki-pages satisfying certain conditions. Also, this extension makes it possible to import several bio-ontologies for limited usage and to export Wiki-content into OWL representation [7]. As the ability of using external ontologies is rather limited, BOWiki cannot be used as a tool for building an arbitrary Wiki-system based on ontology.

There exist MediaWiki extensions working together with Semantic MediaWiki and providing an additional functionality for importing semantic information. One of such extensions is RDFIO [8], which allows importing arbitrary RDF-triplets [9]. For the time being, however, this extension is presented by its beta-version and is not ready to widespread use yet. Another similar extension is LinkedWiki [10], which can be applied to retrieve data from an external service and map them into a Wiki-system. The extension does not allow modifying the structure of the existing system, so its application field is quite limited.

An inverse operation for building a Wiki-system on the basis of subject domain ontology is the operation of knowledge extraction from a Wiki-system. This can be achieved in different ways. Usually, knowledge is extracted in the form of ontology and the main source of this action is Wikipedia. Here are some examples of the implementation of such approaches. The ontological knowledge base YAGO [11] and its extended version YAGO2 [12] use Wikipedia and WordNet [13] for ontology building. These systems extract leaf categories, individuals and most relations from Wikipedia category hierarchy and infoboxes. The remaining part of the hierarchy of categories and relations is completed using the relations and concept

hierarchy of the WordNet. In the project of designing a large-scale person ontology [14], the category hierarchy and individuals are extracted from Wikipedia using a machine learning classifier and the Japanese thesaurus Nihongo Goi-Taikai. There is also a project intended to design subject-oriented ontologies using Wikipedia as the development environment [15]. The authors of this project assume that “the URIs of Wikipedia entries are surprisingly reliable identifiers for ontology concepts”.

In addition to the direct extraction of knowledge in the form of ontology, Wikipedia helps to solve other auxiliary tasks. For example, the text corpus related to a certain subject domain is built from Wikipedia articles [16] and then used for designing the ontology of this subject domain. In the project “Building and Using Geospatial Ontology in the BioCaster Surveillance System” [17], Wikipedia is a data source used to design geospatial ontology. The ontology is built in the semi-automatic mode with automatic data extraction and human-aided verification. The ontology design on the basis of Wikipedia is used as one of stages in another project aimed at an automatic construction of large-scale multi-modality ontology for web image classification [18]. It employs both structural and content features of Wikipedia for the text part and formalizes real world objects in terms of concepts and relationships.

To summarize, all the approaches considered allow extracting some information from Wikipedia for various purposes but none of them can be used as a tool for maintaining a Wiki-system already constructed.

3. Main components of Wiki-systems and tools for working with them

Any Wiki-system is based on a Wiki-engine that is a collection of software utilities for the transformation of Wiki-markup into a code designed to display in the browser. One of most commonly used engines is MediaWiki (widely known Wikipedia is based on it). There exist special additions to MediaWiki called extensions, which provide a certain functionality. For example, the extension Semantic MediaWiki makes it possible to add semantic information through extending the markup and provides facilities to manage such information.

To perform technical tasks in Wiki-systems bots are used. The bot is a software application that runs automated tasks over the Internet. As rule, bots perform tasks that are both simple and structurally repetitive. For example, in Wikipedia bots are used to rename the sets of categories and articles, place interwiki links (links to family projects), correct links, and delete spam. Bots can be implemented in different programming languages, and there exist different libraries for the simplification of their development. One of most advanced library is Python WikipediaBot Framework [19]. It

uses MediaWiki API (special application programming interface) to interact with MediaWiki-system, in particular for authorization, data retrieving and updating.

4. Ontology-based constructing of a Wiki-system

An instrumental system enabling to construct a Wiki-system based on a previously developed ontology was implemented. The ontology defines the structure and content of the Wiki-system constructed. In the future, the constructed Wiki-system can be extended as an ordinary Wiki-system using traditional Wiki-technology tools.

4.1. System structure and operation

A general schema for constructing a Wiki-system involving the instrumental system developed within the framework of the proposed approach combining Wiki-technology and ontology is presented in Figure 1. At the first stage, a qualified knowledge engineer designs a subject domain ontology. At the second stage, a Wiki-interface is constructed using the Onto2Wiki module developed by the authors of this paper. This interface consists of two parts: the designer part and the user part. The former is intended to update the data in the system by subject domain experts. At the same time, all the changes made become available to the users through the user part.

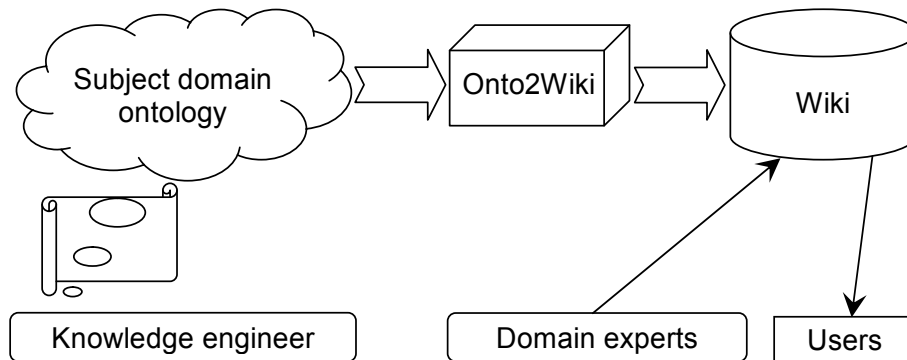


Figure 1. A general schema for constructing a Wiki-system

A detailed schema of the instrumental system operation is presented in Figure 2. According to it, the development of a Wiki-system has the following stages.

At the first stage, the ontology given beforehand or designed in an ontology editor, such as Protégé editor [20], is saved or converted to the OWL format [7]. After that, the file with the ontology specification is parsed by the Onto2Wiki module that uses the RDFLib library [21]. Thereafter,

the Onto2Wiki module constructs an information system framework on the basis of an empty Wiki-site using the Python WikipediaBot Framework. The developed Wiki-site is based on the MediaWiki engine with the Semantic MediaWiki extension. After that, necessary pages are added to the Wiki-system, and attributes, categories and links related to these pages are assigned. When these operations have been completed, the Wiki-system is ready to use.

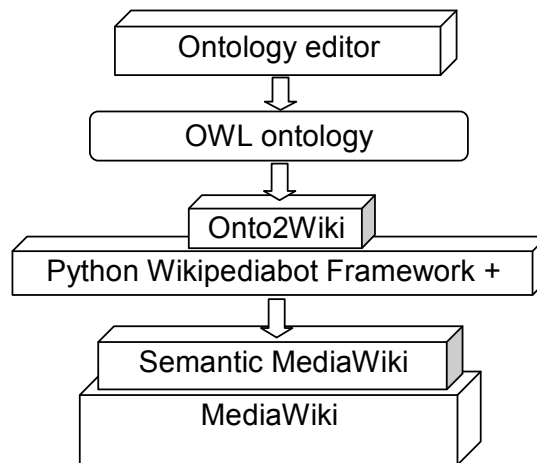


Figure 2. Detailed schema of the instrumental system operation

The correspondence between OWL and Semantic MediaWiki constructions used to map the ontology to the structure and content of a Wiki-system are presented in Table 1. According to the table, a specific Wiki category will be created for every class of an ontology. The Wiki subcategories will correspond to ontology subclasses according to the ontology class hierarchy. A specific Wiki page will be created for every instance of ontology class, and the Wiki page categories will be assigned according to the classes of these instances. Also, all attributes of an ontology instance will be assigned as attributes of the page corresponding to the instance. In addition, all relations of an ontology will be mapped to typed links [4] between the corresponding pages and categories.

If needed, the Wiki-system constructed with help of the Onto2Wiki module can be modified by means of interacting with it as with an ordinary Wiki-system. For example, it is possible to create new pages, assign new links and categories, and so on.

Thus the feature of the proposed approach is the complete independence of the constructed Wiki-system from the ontology on which it is based, i.e., a modification of the Wiki-system does not affect the source ontology. Therefore, a special tool is needed to control the modification of ontology which implicitly forms the bulk of the Wiki-system.

Table 1. Correspondence between OWL and Semantic MediaWiki constructions

OWL	Semantic MediaWiki
Class	Category
Subclass	Subcategory
Instance	Page
Attribute	Attribute
Relation	Typed link

4.2. Example of the system operation

Let us consider the instrumental system operation by the example of constructing the teaching information system “Technology of creating expert systems”. Consider a fragment of the constructed ontology in Turtle notation:

```
ex:Build_Your_Own_ES rdf:type ex:Literature_about_ES ,
                        owl:NamedIndividual ;
ex:Book_title "Build Your Own Expert System" ;
ex:Book_author "Chris Naylor" ;
ex:Refer_to ex:DENDRAL ,
            ex:Expert_systems ,
            ex:MYCIN ,
            ex:PROSPECTOR ,
            ex:PUFF .
```

We input the ontology into the system and after the completion of its operation get a constructed Wiki-system.

An example of one page of the system derived is presented in Figure 3. On the page, we can see attributes and typed links.

And here is Wiki-markup of this page:

Individual Build Your Own ES

Book title: [[Book title::Build Your Own Expert System]]

Book author: [[Book author::Chris Naylor]]

Refer to: [[Refer to::DENDRAL]]

Refer to: [[Refer to::Expert systems]]

Refer to: [[Refer to::MYCIN]]

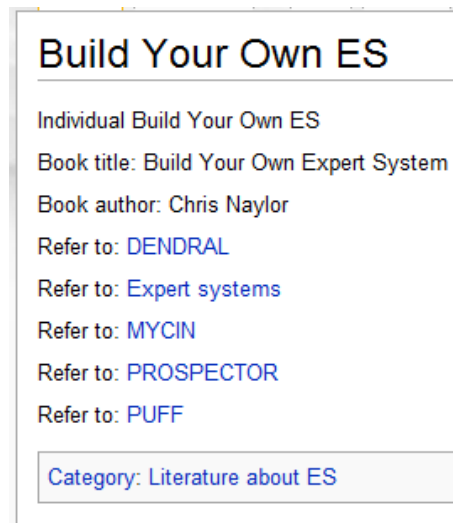


Figure 3. An example of a page

Refer to: [[Refer to::PROSPECTOR]]

Refer to: [[Refer to::PUFF]]

[[Category:Literature about ES]]

5. Extracting ontologies from Wiki-systems

As discussed above, the approach to the construction of a Wiki-system based on a subject domain ontology should include, in addition to mapping an ontology to the structure and to the content of a Wiki-system, the inverse procedure, i.e. the method of extracting ontologies from Wiki-systems. This method could be applied both to maintain the information systems constructed and to facilitate the development of a new Wiki-system on the basis of several Wiki-systems.

5.1. Ontology extraction method

The task of extracting an ontology from a Wiki-system is solved with the tools applied to mapping. The only difference is that the Wiki2Onto module specifically developed under this approach is used instead of Onto2Wiki module. All operations are executed in the inverse order. The Wiki2Onto module extracts an ontology from a Wiki-system using the Python WikipediaBot Framework. Then it saves the ontology to the OWL file using RDFLib. The same table of correspondence between OWL and semantic Wiki con-

structions is used (see Table 1).

Let us consider this process in more detail. An ontology is extracted from a Wiki-system in the following order. First, all the classes are extracted: each Wiki-category is mapped to one class; a hierarchy of ontology classes is defined by the structure of the nesting Wiki-categories. Then each Wiki-page is presented as an instance of the class corresponding to the Wiki-page category. For empty pages having incoming links in a Wiki-system, a special auxiliary class “Non-existent pages” is created.

After that, all the links of each page are processed. First of all, it is determined whether a link is ordinary or semantic. If the link is ordinary, the relation “Link to” is created which links the instance corresponding to the processed page with the instance whose name coincides with the name of the page at which the link points. If a link is semantic, (it is also called a “property” in the Semantic MediaWiki system), it has a structure that looks like $\langle \textit{property name}, \textit{property value} \rangle$. The type of the property is analyzed. If the property has the type “Page” or its type is not specified, then the relation with the corresponding name (*property name*) and value (*property value*) is created. (Note that by default the link property has the type “Page”.) If the property has another standard type, then an attribute whose type is determined according to Table 2 is created for the instance. For user property types, the types of attributes are defined by the names of these types.

Table 2. Correspondence of standard types under extraction

SMW property type	OWL type
String	string
Number	double
Boolean	boolean
Date	dateTime
Text	string
Code	string

Not that only the standard Semantic MediaWiki property types presented in Table 2 are supported in the current realization of the Wiki2Onto module.

Also, note that a Wiki-system from which an ontology is extracted does not necessarily have a Semantic MediaWiki extension. However, if it does not have such an extension, the extracted ontology will be poorer because it will lack specific semantic information. In particular, attributes cannot be extracted, and the variety of relations will be insignificant. At the same time, the amount of information extracted can be enlarged by adjustment to a concrete Wiki-system.

It goes without saying that the ontology extracted can be inconsistent and can contain duplicated information. The essence of the suggested approach is that it is much simpler to verify an ontology than a Wiki-system. Such verification shows places in the source wiki-system which have problems. This and other possible applications of the approach are described in the next subsection.

5.2. Applications of the ontology extracting method

The main application of the ontology extracting method is tracing the modifications of the source ontology in process of the evolution of a Wiki-system based on it. It can be useful not only for the sake of curiosity but also for more significant goals such as tracking the project evolution, verification of a derivable ontology, control quality and balance of the resulting data structure, and coordination of the evolution of different parts of a subject domain. Also the extracted ontology can be processed by one of existing reasoners to derive implicit knowledge.

Besides, an ontology can be extracted not only from the Wiki-system built on this ontology but also from any other Wiki-system. For example, it is necessary to design an ontology of some subject domain and there exists a Wiki-system that contains the necessary information from this domain. Instead of manually designing the required ontology from scratch, we can extract a preliminary draft ontology from the Wiki-system and then rework it for our goals (correcting the structure, removing unnecessary parts and adding missing details). This way of ontology building is much easier.

Another application of the method is merging several Wiki-systems related to close subject domains. A direct merging of Wiki-systems could be rather difficult and take very much manual work, since it is extremely hard to find all links and intersections between two systems. It could be made much easier using the approach suggested: extract an ontology from each Wiki-system, merge these ontologies (this issue is beyond the scope of this article), and then map the ontology obtained to the resulting Wiki-system.

5.3. Ontology extracting example

Let us consider an example of applying the ontology extraction method to retrieve an ontology from a Wiki-system. Let us have a Wiki-system with two informative pages and several service pages. The first informative page is “Main page”. Its code in Wiki-markup looks like as this:

```
[[Individual::Person|Article about person]]
```

The second informative page is titled “Person”. Its code in Wiki-markup is

His name is [[Name::Edward]].

Number of sons: [[Number of sons::2]]

He [[Married::false|is not married]].

Was born [[Date of birth::4 December 1976]]

Characteristic: [[Description:: character is resistant, Nordic]]

Code sample: [[Code fragment::<FORM name="form_name">]]

His number: [[Mobile phone number::+7-955-555-55-55]]

Site: [[Site::http://example.org]]

E-mail: [[Email::edward@example.org]]

See also: [[See also::http://example.com/foaf.rdf|FOAF]]

[[Category:People]]

Let us assume that with the help of service pages the property types are defined as follows: “Name” – String, “Number of sons” – Number, “Married” – Boolean, “Date of birth” – Data, “Description” – Text, “Code fragment” – Code, “Mobile phone number” – Telephone number, “Site” – URL, “Email” – Email, “See also” – Annotation URI.

Then the ontology extracted from this Wiki-system in Turtle notation will be look as follows:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ex: <http://example.org/Ontology100500.owl#> .

<http://example.org/Ontology100500.owl> rdf:type owl:Ontology .
ex:Date_of_birth rdf:type owl:DatatypeProperty .
ex:Married rdf:type owl:DatatypeProperty .
ex:Main_page rdf:type owl:NamedIndividual;
    ex:Individual ex:Person .
ex:Name rdf:type owl:DatatypeProperty .
ex:Individual rdf:type owl:ObjectProperty .
ex:Number_of_sons rdf:type owl:DatatypeProperty .
ex:Non-existent_pages rdf:type owl:Class;
    rdfs:comment "Service class"^^xsd:string .
```

```
ex:Mobile_phone_number rdf:type owl:AnnotationProperty,  
                        owl:DatatypeProperty .  
ex:Description rdf:type owl:DatatypeProperty .  
ex:Code_fragment rdf:type owl:DatatypeProperty .  
ex:Site rdf:type owl:AnnotationProperty,  
        owl:DatatypeProperty .  
ex:See_also rdf:type owl:AnnotationProperty,  
            owl:DatatypeProperty .  
ex:Email rdf:type owl:AnnotationProperty,  
         owl:DatatypeProperty .  
ex:People rdf:type owl:Class .  
ex:Person ex:Date_of_birth "1976-12-04T00:00:00"^^xsd:dateTime;  
         ex:Married "false"^^xsd:boolean;  
         ex:Name "Edward"^^xsd:string;  
         ex:Number_of_sons "2"^^xsd:double;  
         ex:Mobile_phone_number <tel:+7-955-555-55-55>;  
         ex:Description "character is resistant, Nordic"^^xsd:string;  
         ex:Code_fragment "<FORM name=\"form name\">"^^xsd:string;  
         ex:Site <http://example.org>;  
         ex:See_also <http://example.com/foaf.rdf>;  
         ex:Email <mailto:edward@example.org>;  
         rdf:type ex:People,  
                owl:NamedIndividual .
```

6. Conclusions

An approach to constructing information systems based on the Wiki-technology and subject domain ontology was considered. Within the framework of this approach, two methods have been proposed: ontology-based creation of Wiki-systems and ontology extraction from the existing Wiki-systems. Also, a prototype of the tool capable of realizing these two methods to the extent specified in Subsections 4.1 and 5.1 has been developed. The method developed for the ontology-based creation of Wiki-systems allows us to build a Wiki-system with a good structure and a consistent concept system.

The method of ontology extraction from the existing Wiki-systems opens up the following possibilities for building information systems and ontologies.

1. **Quality control of a Wiki-system constructed on the basis of an ontology throughout its life-cycle.** A Wiki-system built within the framework of the approach proposed is supplemented with new categories, pages and links; from time to time, the ontology is extracted from this system and verified, and necessary corrections are made in the source Wiki-system; as a result, the quality of the Wiki-system is improving.

2. **Reengineering a Wiki-system.** First, an ontology is extracted from a Wiki-system, verified and supplemented with required elements (if necessary). After that, a consistent Wiki-system possessing new properties is built based on newly obtained ontology.
3. **Building a generalized Wiki-system on the basis of several Wiki-systems related to similar topics.** From each of Wiki-system an ontology is extracted. The ontologies extracted are merged into one. The ontology obtained is used to build a generalized Wiki-system.
4. **Building the sufficiently complete ontology of subject domain.** Ontologies are extracted from several Wiki-systems related to the same subject domain. After the verification of the ontologies obtained, they are united in one ontology (using the methods of ontology alignment and merging).

References

- [1] Gavrilova T., Gorovoy V., Petrashen E. Learning Resources Organization Using Ontological Framework // *Advances in Web Based Learning – ICWL 2009*. – Springer, 2009. – P. 158–161.
- [2] MediaWiki. – <http://mediawiki.org>.
- [3] Semantic MediaWiki. – <http://semantic-mediawiki.org>.
- [4] Völkel M., Krötzsch M., Vrandečić D., Haller H., Studer R. Semantic Wikipedia // *Proc. of the 15th Internat. Conf. on World Wide Web (Edinburgh, Scotland, May 23–26, 2006)*. WWW '06. – NY: ACM Press, 2006. – P. 585–594.
- [5] Semantic MediaWiki ontology import. – http://semantic-mediawiki.org/wiki/Help:Ontology_import.
- [6] Backhaus M., Kelso J., Bacher J., Herre H., Hoehndorf R., Loebe F., Visagie J. BOWiki – a collaborative annotation and ontology curation framework // *Proc. of the Workshop on Social and Collaborative Construction of Structured Knowledge, CKC Banff, Canada, May 8, of CEUR Workshop Proceedings*. / Noy N., Alani H., Stumme G., Mika P., Sure Y., Vrandečić D. (Eds). – Aachen: CEUR-WS.org, 2007. – Vol. 273.
- [7] OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax / Ed. by Motik B., Patel-Schneider P. F., Parsia B. – W3C Recommendation, 27 October 2009. – <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>.
- [8] RDFIO. – <http://mediawiki.org/wiki/Extension:RDFIO>.
- [9] RDF. – <http://w3.org/RDF/>.

-
- [10] LinkedWiki. – <http://mediawiki.org/wiki/Extension:LinkedWiki>.
 - [11] Suchanek F. M., Kasneci G., Weikum G. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia // Proc. of the 16th Internat. Conf. on World Wide Web (Banff, Alberta, Canada, May 8–12, 2007). WWW '07. – NY: ACM Press, 2007. – P. 697–706.
 - [12] Hoffart J., Suchanek F. M., Berberich K., Weikum G. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. – November 2010. – (Research Rep. / Max-Planck-Institut für Informatik; MPI-I-2010-5-007).
 - [13] WordNet: An Electronic Lexical Database / Fellbaum C. (Ed.). – Cambridge, MA: MIT Press, 1998.
 - [14] Shibaki Y., Nagata M., Yamamoto K. Constructing large-scale person ontology from Wikipedia // Proc. of the 2nd Workshop on “Collaboratively Constructed Semantic Resources”. – Coling, 2010. – P. 1–9.
 - [15] Hepp M., Bachlechner D., Siorpaes K. Harvesting Wiki consensus – using Wikipedia entries as ontology elements // Proc. of the First Workshop on Semantic Wikis – From Wiki to Semantics, co-located with the 3rd Annual European Semantic Web Conference (ESWC 2006). – 2006. – P. 124–138.
 - [16] Cui G. Y., Lu Q., Li W. J., Chen Y. R. Corpus exploitation from Wikipedia for ontology construction // Proc. of the Sixth Internat. Language Resources and Evaluation (LREC 2008). – Marrakech, 2008. – P. 2125–2132.
 - [17] Doan S., Ngo Q.-H., Kawazoe A., Collier N. Building and Using Geospatial Ontology in the BioCaster Surveillance System. – Available from Nature Precedings, <<http://dx.doi.org/10.1038/npre.2008.2110.1>> (2008).
 - [18] Wang H., Jiang X., Chia L.-T., Tan A.-H. Wikipedia2Onto — Adding Wikipedia Semantics to Web Image Retrieval // Proc. of the WebSci'09: Society On-Line, 18–20 March 2009, Athens, Greece.
 - [19] Python WikipediaBot Framework. – <http://pywikipediabot.sourceforge.net>.
 - [20] Protégé. – <http://protege.stanford.edu>.
 - [21] RDFLib. – <http://rdflib.net>.

