

Expressive Capabilities of Semantic MediaWiki: Advantages and Limitations

Julia Rogushina¹

¹ Institute of Software Systems of the National Academy of Sciences of Ukraine, 40, Ave Glushkov, Kyiv, 03181, Ukraine

Abstract

We consider basic functional components of semantic search, the criteria for evaluating search languages and classification of search engines to define this umbrella concept for specifics of resources based on wiki technologies. Possibilities of semantic search are based on expressiveness of queries that use semantic properties of information objects represented into wiki resources. of semantic structuring of resource content are analyzed.

We analyze additional opportunities that the use of the Semantic MediaWiki plug-in provides for the resources built on the MediaWiki technological platform for building semantic queries.

Semantization of already existing wiki resources differs from the development of semantic ones, and we compare main steps of these processes and advantages of use the ontological model in them. This model provides an unambiguous interpretation of the relations between typical information objects represented into the resource, their properties and restrictions. Proposed approaches to semantization are tested on three independent information resources of different types that use the wiki technological platform for collaborative processing of distributed data and knowledge. They can be useful for making decisions about the expediency of semantization of information resources with different scope and purposes and for determining the most effective ways of implementing the chosen solution.

Keywords 1

Wiki technologies, Semantic MediaWiki, semantic search, ontology

1. Semantic search

Semantic search (SS) is an umbrella term that is used to denote a group of models and methods using external knowledge sources that improve traditional search approaches in various ways, using the context and semantics of both the user's query and information resources (IRs) in where this search is carried out.

- Search capabilities in the most general form are determined by [1]:
- means of describing the users request that represent their information need;
- means of description and structuring of the data set where this search is carried out;
- methods of matching of user request with data elements;
- external and internal knowledge used for semantic processing of requests, for semantic structuring of data and for describing of user sphere of interests;
- methods of the search result representation.

SS is one of the components of the IR semantization, which also includes means of semantic structuring of content, navigation instruments, metadata generation and representation, knowledge import and export tools, content consistency checks, etc.

- Two groups of SS functional components can be distinguished [2]:
- improvement of knowledge-oriented processing of initial user request;
- semantic structuring of the content and metadata of the data set.

14th International Scientific and Practical Conference from Programming UkrPROG'2024, May 14-15, 2024, Kyiv, Ukraine

* Corresponding author.

✉ ladamandraka2010@gmail.com (J. Rogushina)

📄 0000-0001-7958-2557 (J. Rogushina)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Quite often, IR developers propose SS support for customers, but developers and customers can understand functionality of SS completely differently by:

- methods used for descriptions of user needs;
- types of external sources of knowledge and ways of their selection and use;
- structure of retrieved information objects and their components;
- forms of search result representation, their possible properties and values.
-

Such ambiguity is caused by fuzzy definitions of SS concept and complicates mutual understanding of SS possibilities and goals in particular applications. As a result, developers create some product that is not sufficient for customer needs, and the already chosen technological platform does not allow making the necessary improvements. Therefore, it is important to define clearly what kind of SS provides some technological solution and what efforts of developers are required to use semantics for some IR on base on this solution.

Analyses of the search language has to determine:

- what parameters can be used in the query conditions;
- what types of values of these parameters are supported;
- what operations between these values (comparison, logical, arithmetic, etc.) are supported.

The choice of specific models and tools depends on the purpose of IR development and on capabilities available for users of such resource. But effectiveness of this choice is defined by analytical reviews of individual solutions, based on practical experience of their application for representation of content that differs by volumes, dynamics and heterogeneity.

Need in use of practical experience of IR developers in these reviews is explained by the fact that some capabilities declared in such technological solutions are too complex for users, inconvenient or slow to be used in scaling applications. It is also important to consider the differences of possibilities between software versions because they can significantly affect the results.

Many researchers analyze the expressiveness of query languages for semantic resources [3, 4] analyzes languages used for information structuring and semantic markup such as XML, RDF and DAML+OIL with the corresponding ontology schemas and specifications. The purpose of such research is to determine criteria for evaluating the expressiveness of a markup language that can be used for its choice for practical tasks.

With use of this recommendations, we consider the following criteria for comparison of markup languages that are based on elements of query conditions:

- Subclasses and properties: what relations between classes of object (both “class-subclass” and task-specific ones), between classes and instances of classes, between instances of classes and their properties markup language allows to define;
- Atomic data types: what data types (such as string or number) can be used to describe the data;
- Instances: how instances of classes can be described (in terms of properties, belonging to classes, constraints, etc.);
- Property Constraints: what is a complexity of property constraints can be defined on classes and class instances (such as domain, range, range power, mandatory value of property, etc.);
- Property values: is it possible to specify default values, valid and invalid values;
- Context: how the markup language reflects different contexts (e.g. namespaces) of interpretation;
- Support for logical operations: does the language allow the use of negation, conjunction and disjunction operators to describe relations between classes and instances of classes;

- Inheritance: what restrictions and property values of parent classes can be propagated to subclasses.

In addition to these criteria, it is advisable to take into account the convenience of practical using the markup language for IR structuring, the availability of editing tools and means for analysis of input errors.

Many practical tasks require a limited subset of these features to satisfy the informational needs of users, and then the choice of markup language is based on its usability and availability of automated error control for markup creation.

Many researchers analyze the expressiveness of query languages for semantic resources such as the SPARQL language for searching into RDF and OWL. SPARQL has high expressiveness and ensures high pertinence of search results. Unfortunately, such resources based on formal knowledge representation now represent only a small part of the Web content, and constructing SPARQL queries is rather complex. This fact causes the need in methods that support SS into semi-structures and non-structures resources – by additional transformation of queries and data.

Structured semantic queries based on ontologies can use elements of domain ontology such as class concepts, class instances and their properties. At the same time, the expressiveness of the ontology-based request depends on types of IR characteristics that can be used into this query. Main types of such characteristics are:

- anonymous relations between IR content elements where the request ignores the name and semantics of this relation and takes into account only the existence of such relation (an example is hyperlinks in the web documents used by Google search);
- usual properties that are associated with logical relations between content elements (for example, synonymy, "class-subclass" relation, "category instance", html markup elements);
- domain-specific arbitrary relations that can be defined as properties of instances of domain-specific objects (for example, object relations of organization ontology "work in an institution", "have a position" or e-library relation "author of the publication").

The SS possibilities are supplemented by the use of external sources of knowledge and methods of their application in the search process. One of them is the creation and analysis of semantic markup of information resources. Therefore, significant attention of researchers in this direction is paid to the creation of semantic markup languages and the comparison of their expressive capabilities. SS can use an ontology as a source of domain concepts, their structural elements and their possible values. User can describe in request the set of desirable and undesirable values of retrieved objects, define their type, etc. For example, user can select objects from class "Organization" with values "Lviv" or "Kyiv" of the property "Location".

Many approaches to semantic retrieval are based on the Semantic Web but differ significantly by architecture, user content processing, query representation, etc. One of classification criteria set for SS is proposed in [5].

The expressiveness of SS depends significantly on the pertinence between the set of documents or other information objects where this search is carried out, and the domain ontologies. If ontology is selected correctly, then the metadata of the documents clearly refer to the concepts of a specific ontology and vice versa. Sometimes such objects are considered as separate instances in the ontology. Using this approach, it is easy to resolve homonymy and refine queries, but it causes more complex creating a semantic document annotation.

Another important factor of semantic search is its transparency that characterizes the user's interaction with search functions. Transparent systems, where semantic capabilities are invisible to the user, have no means to get additional information from the user, for example, for clarification of homonyms or selection of an external knowledge base. Interactive systems allow to receive request

clarification from the user or recommend request changes. Hybrid systems combine interactive and transparent behavior – they usually act as transparent ones and require user interaction only for some tasks. Transparent systems are easier to use, but the user cannot influence the system's semantic decisions, therefore potential quality of their search results is reducing.

It is worth to note, that the usefulness of SS results also depends on the user's personal settings based on context processing. Examples of the search context use are:

- machine learning based on the history of interaction with the user (in this system or in others);
- explicit determining the desired categories of retrieved information objects by the user (from some ontology or taxonomy);
- individual selection of knowledge base used for search;
- use of experience of interaction with users that have similar information needs (recommending systems).

2. Semantic search into Wiki resources

Currently, many IRs are based on wiki technologies. Such resources are represented by collection of pages with unique identifiers that can contain natural language text, multimedia elements (pictures, videos, audio files, etc.) and some elements of Wiki markup that define relations between these pages and provide the basis for knowledge sharing. Wiki content is oriented both on human usage and automated processing. Wiki technologies are oriented on collaborative development of content, mutual work of big groups of users and representation of large volumes of data.

Wide use of wiki technology is caused by:

- their relative simplicity for the end user;
- support for collaborative work with content;
- ability to scaling for large amounts of information.

Additional potential of wiki-based IRs deals with possibilities of semantic markup where relevant domain concepts (for example, from domain ontologies or thesauri selected according to user needs) are used as tags for content structuring. Such structures IRs provide a more convenient means of information retrieval where user requests can be represented in domain concepts.

Wikis have a large number of software solutions for their semantization that provide additional means of content search, view and structuring [1, 2]. These solutions differ significantly by the possibilities provided by such semantization. Many semantic Wikis use ontologies to describe the knowledge base of IR, user profiling, data modeling, etc.. Some of them provide users with an interface to create ontologies and to execute SPARQL queries, others offer their own search languages.

The expressiveness of SS in wiki resources depends on:

- semantic markup elements they can be used in queries;
- complexity of the markup language;
- usability of search constructing.

These software solutions propose various powerful query languages that offer a variety of possibilities for SS, but the syntax of formal query languages is rather complex for end users. Therefore, the challenge arises to find such approaches to semantic search that combine the expressiveness and capabilities of structured queries with the simplicity of traditional keyword searches. For example, in [6] SS is implemented as an extension of traditional search: users formulate their information needs by the set of keywords, and this set is transformed into structured query that can be further clarified by the user.

3. Task definition

The aim of this research is to analyze additional possibilities for search and navigation provided by the Semantic MediaWiki (SMW) plug-in for MediaWiki technological environment. Other question deals with additional efforts of IR developers required for implementation of such semantization in order to determine its feasibility for practical task.

For this purpose, the following questions are investigated in the work:

- what additional possibilities does the use of the SMW semantic plug-in provide;
- what SS elements can expand the functionality of wiki resource without semantic markup only by installing SMW;
- how templates can be used to transition from a non-semantic wiki resource to the semantic one;
- how SMW templates differ from traditional wiki ones;
- what additional elements should be developed for support the knowledge base of the semantic wiki resource;
- what problems can semantization of a wiki resource cause and what should be done to eliminate them.

Such an analysis should become the basis for choice of the suitable technological platform for development of Wiki resources with support of semantic search functionality that satisfies information needs of IR customers.

4. Semantic search for Wiki resources

Now a lot of Wiki resources (such as Wikipedia) use MediaWiki technological platform [7]. The main structuring mechanism of MediaWiki is based on wiki pages and their categories, and SMW semantic plug-in [] expands it by additional means.

SS based on the SMW works on the basis of explicit structuring of content with an arbitrary set of markup tags. The main data structuring primitives in SMW assume a formal semantic interpretation in terms of ontological analysis for OWL DL. Each page can be assigned to one or more categories, and these categories can be linked by hierarchical relations.

SMW provides ways to add additional structure to MediaWiki through the semantic markup of wiki content: semantic properties of the wiki page represent binary relations between this page and other entities such as wiki pages or data values. Meaning of every such relation is defined by appropriate markup tag. These tags can be extracted from the ontology of the relevant domain that formalize its semantic interpretation or selected by IR developers according to their goals.

4.1 Ontological model of semantic wiki resource

If, as a result of the semantization of Wiki resource, its knowledge base becomes quite complex, then we require means to formalize its characteristics in some interoperative representation. For example, we can use an ontology of the relevant domain. Such ontology captures the semantics of the connections between the types of information objects and their templates, the semantic properties of these objects, their categories, etc. Ontological representation provides an unambiguous interpretation of this information, and the availability of commonly accepted formats (OWL, RDF) and convenient tools for working with them (such as Protégé) simplifies interaction with users and other systems, and also supports the reuse of information. Visualization of the necessary fragments of such an ontology (Figure 1) helps users to work with Wiki templates and understand meanings of their components, but it is necessary to maintain the synchronization of IR ontology with current changes in its knowledge base.

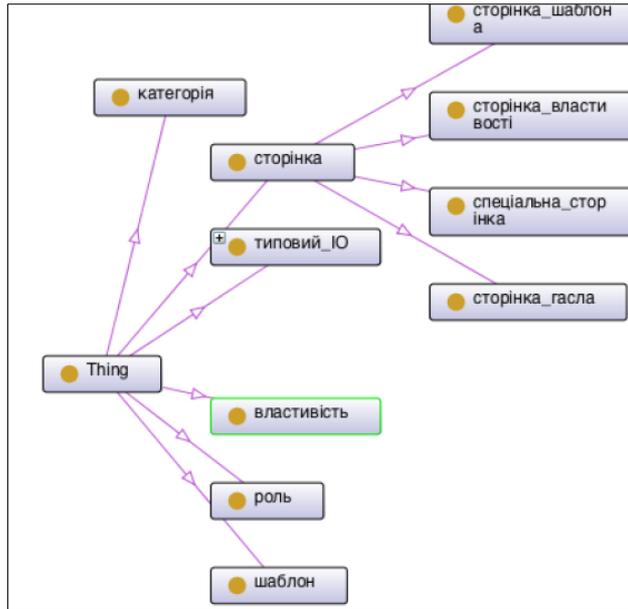


Figure 1: Ontological model of semantic Wiki resource (fragment).

The formal semantics of structured data in SMW can be provided through mapping to the OWL ontology language [9].

We can use the unambiguous correspondences:

- Regular article wiki pages correspond to instances of OWL ontology classes;
- Wiki categories correspond to classes;
- SMW semantic properties correspond to properties (SMW properties with values of type "Page" map to ontology object properties, and properties with other data types map to ontology data properties).

This model formalizes information about IR objects and provides the semantics of its elements without direct contact with its developers.

Accordingly, property values can be ontology instances or constants. Categories of wiki pages define their class in OWL. MediaWiki supports a hierarchical organization of categories, and SMW can interpret this set of categories as hierarchy of OWL classes.

Ontological representation of non-semantic Wiki resource $O_{wiki} = \langle P = P_{user} \cup P_{categ} \cup P_{spec}, L = \{"link"\} \rangle$ contains the following elements:

- the set of Wiki pages $P = P_{user} \cup P_{categ} \cup P_{template} \cup P_{spec}$ where P_{user} is a set of user papers, P_{categ} is a set of pages that define categories, $P_{template}$ is a set of pages that define templates, P_{spec} is a set of other special pages;
- $L = \{"link"\}$ is a one-element set that defines relation "link from current page to another one".

Formal model of the semantic Wiki resource includes additional components that describe semantic properties of wiki pages [10]: $W_s = \langle P, L = \{"link"\} \cup L_{sem_prop} \rangle$, where set of the wiki pages $P = P_{user} \cup P_{categ} \cup P_{template} \cup P_{sem_prop} \cup P_{spec}$ is enriched by P_{sem_prop} that defines semantic properties of Wiki pages, where some properties define relation of current page with other

ones: $P_{sem_prop_page} \subseteq P_{sem_prop}$, and other relations link current page with values of selected data

type: $L_{sem_prop} = \{l_i\}, i = \overline{1, n}$.

The main advantage of SMW-based search – in contrast to traditional wiki searches by categories – is a simultaneous use of the set of requirements for categories and values of semantic properties of wiki page into one query. Thus, even without semantic markup of IR content, we can use queries with a set of categories.

4.2 SMW search language

SMW proposes the ASK language for representation of structured queries. ASK allows to define:

- restrictions on the set of categories and values of semantic properties of wiki pages that are interesting for user;
- order of result representation;
- set of displayed semantic properties of retrieved pages;
- the number of results proposed to users;
- format for result representation.

SMW queries allow to display not the entire content of pages or their identifiers, but user can define the set of properties and receives their values. In addition, queries allows to define the form of result representation – table, list, diagram, gallery, etc., and to limit the number of results proposed to user.

Constraints of ASK query allow to compare property values with constants of different types, but do not support performing complex calculations. They can contain comparators for describing of matching type. Comparators are special characters that are placed in query after “:” between property value and selected constant: for example, [[Year::>>1930]], [[Organization::!~National Academy of Sciences of Ukraine]].

The following comparators are supported in SMW [Search operators. – www.semantic-mediawiki.org/wiki/Help:Search_operators]:

- “<” – “more”;
- “>” – “less”;
- “!” – “not equal to”;
- “>>” – “greater or equal to”;
- “<<<” – “less or equal to”;
- “~” – “string matches”;
- “!~” – “string does not match”.

Correct use of such comparators in conditions requires to define correctly the type of the property (property type is defined when the semantic property is created but can be changed later), because such conditions have different results for different comparison of values of different type. for example, for properties of “Integer” type the value 1020 is greater than 105, and for properties of “String” type the value 1020 is less than 105.

It is important to clearly understand the meaning of comparators in SMW query conditions. For example, the condition [[Birthplace::!Lviv]] allows to select pages with “Birthplace” property values that differ from “Lviv”. This condition does not look for pages that don’t have any value of “Birthplace” property, but instead it selects pages that have a value for this property and this value is not “Lviv”.

The use of strict comparators “<” and “>” can lead to incorrect interpretation caused by comparing values in different measurement units due to different rounding options (defined by the

administrator using the “\$smwStrictComparators” configuration parameter). Therefore, it is more useful to use condition pairs with property values in a certain range specified by the nonrigorous comparators “<” and “>”, for example: `[[Height::>4 feet]]` `[[Height::<10 feet]]` . Comparators can also be applied to page names (without a namespace prefix).

Also, such a condition can be described using a logical disjunctive relation, denoted by the symbols “|”, for example: `[[height::>4 feet|<10 feet]]`. Wildcards (“+” – any value, “*” – an arbitrary sequence of characters and “?” – any single character) expand the possibilities of describing the values of semantic properties in queries.

There are three types of “magic words” in MediaWiki [Help:Magic words. – www.mediawiki.org/wiki/Help:Magic_words]:

- *behavior switches* that control the behavior of pages and the representation of information on them;
- *variables* that return information about the current page, time and environment;
- *analyzer functions* (parser).

Queries can also contain MediaWiki's magic words that return information about current page, time, environment and arbitrary wiki pages instead of constants .

Variables are written as strings of uppercase characters separated by double curly braces, similar to wiki templates: `{{FOO}}`. They allow to receive information in different representation – for example, the current month can be indicated by a number or a name. The most commonly used variables in MediaWiki are:

- `{{CURRENTYEAR}}` – current year;
- `{{CURRENTMONTH}}` – current month;
- `{{CURRENTDAY}}` – current day of the month;
- `{{CURRENTDOW}}` – current day of the week;
- `{{CURRENTTIME}}` – current time (in 24-hour format).

Other variables provide access to technical metadata and wiki page parameters. For example, `{{SITENAME}}` returns the name of the wiki resource site, `{{SERVERNAME}}` returns the name of the server where it is located, and `{{CURRENTVERSION}}` returns the current version of MediaWiki. The variables `{{REVISIONDAY}}`, `{{REVISIONMONTH}}` and `{{REVISIONYEAR}}` return information about the day, month and year of the last revision of the page, and `{{REVISIONUSER}}` – information about the user who made this revision.

Other group of variables returns elements the wiki resource statistics. For example, `{{NUMBEROFPAGES}}` returns the total number of wiki pages, `{{NUMBEROFARTICLES}}` – the number of wiki pages in the content namespace, `{{NUMBEROFUSERS}}` – the number of users, `{{NUMBEROFACTIVEUSERS}}` – the number of active users, `{{PAGESINNS:index}}` – the number of wiki pages in the selected namespace.

The `{{PAGENAME}}` variable returns the name of the current wiki page. IR developers have to take into account that the set of magic words with these characteristics is defined for MediaWiki, and some their specifics depend on its version. Representation of information returned by from variables depends on the skin and other settings of a specific wiki resource (Figure 1), but not on the presence of the SMW plug-in.

Parser functions can have one or more parameters designated by lowercase letters in double curly braces: `{{foo:...}}` or `{{#foo:...}}`. For example, `{{PAGESIZE:aaa}}` returns the number of characters on the wiki page “aaa”. This extends the possibilities of the variables of the previous group. Other examples of parser functions are `{{REVISIONDAY: aaa}}`, `{{REVISIONMONTH: aaa}}`, and `{{REVISIONYEAR: aaa}}` that allow to get the day, month, and year of the last modification of the

page “aaa”. We can consider variables as parser function with parameter value of current page name.

All these magic words can be used in non-semantic MediaWiki resources, but the use of SS based on SMW and the construction of semantic templates greatly expands the scope of their application and makes such search more flexible because they can be added to query conditions embedded in wiki pages or to explanations of the results of its execution.

4.3 Advantages of SMW use

Analyses of base SMW possibilities allows to distinguish main advantages of its use:

- ability to define explicitly the content of links between wiki pages can be used both for automated content processing and for understanding information by users;
- search by arbitrary combinations of categories and values of semantic properties increases possibilities of the single query;
- extraction of important data from semantic markup of query makes results more understandable and reduces the time of their perception;
- possibility of automated content generation of wiki pages based on built-in queries reduces the time of content development and raises its consistency;
- use of template parameters for generation of semantic markup simplifies this process and reduces the number of input errors.

If some of these advantages are important for developers of the wiki resource, then semantization is reasonable. But they have to take into account that IR semantization requires additional efforts and assumes that they have additional competencies.

Semantization complexity increases non-linearly with an increase of the number of semantic properties and templates that use them. Increasing the number of usual wiki pages affects complexity only linearly: the semantization of each individual page takes approximately the same time but this time increases slightly due to the longer search for correct links to other wiki pages in a longer list of the semantic properties. To ensure the benefits of SMW use, IR developers have to perform the following actions (Table 1).

Table 1

Elements of wiki resource semantics and conditions of their use

Semantic element	Actions required
Creating of semantic properties	create semantic property pages, define their type, and describe their semantics and scope
Defining the set of necessary semantic properties	Discover different properties with the same or similar names , as well as properties with the same semantics but with different names
Manually adding semantic markup to the content of wiki pages	monitor the actual set of semantic properties, understand their meaning and know the rules of semantic markup, as well as analyze the relevance of their application to specific elements of the content
Adding semantic markup by templates	monitor the presence of relevant templates, their current structure and parameter types, know the rules of their use in wiki pages and formatting restrictions in template values
Correct indexing of the resource database	Administration of database indexing rules according to IR specifics and volume
Creation of built-in semantic queries	use the SMW Structured Query Language and analyze the feasibility of adding search results to wiki pages

Ontological model of IR can help in this process by formal representation of knowledge base structure.

4.4 Specifics of IR semantization on various stages

It is important to distinguish the actions performed in the case if the resource is developed as semantic one (that is, all semantic plug-ins are installed before the start of content creation), from the actions executed for semantization of an already existing wiki resource with a large number of pages.

In the first case, the semantization procedure provided simultaneously with development of IR in general consists of following actions:

- development of generalized ontological model of resource;
- defining of typical information objects (TIOs) of this resource and TIO properties;
- defining the types of TIO properties, their possible values, the admissibility of multiplicity and uncertainty;
- creation of pages for corresponding semantic properties ;
- generation of wiki pages where content is marked up by these semantic properties;
- development of wiki templates for TIOs that provide unified input and representation of information;
- testing of TIO templates for real domain instances;
- refinement of ontological model of the IR knowledge base by information about specific features of TIO instances and their relations;
- constructing of semantic queries that obtain information about TIO instances.

In the second case, we start semantization procedure for IR that already contains a lot of non-semantic wiki pages of instances and certain groups of TIO united in categories. Links already exist between pages without defining their semantics. Moreover, some templates are already developed to represent the structure of these TIOs (but structural elements are not formalized by semantic properties), and we have to transform these templates into semantic ones. Therefore, the semantization process includes the following actions:

- install the Semantic MediaWiki plug-in (if necessary, other semantic plug-ins such as Semantic forms);
- analyze the meaning of links between wiki pages, and if a sufficient number of links has the same or similar semantics (what number is considered as sufficient depends on the total volume of the resource and the requirements of the developers) then we create a semantic property of the "Page" type with a pertinent name, and replace the corresponding anonymous links between pages with semantic (these actions are performed for each group of links with similar semantics);
- analyze the existing templates for TIO representation and their parameters, create semantic properties of the corresponding types (it should be noted that by default, after installing SMW, all template parameters are interpreted as properties of the "Page" type, and it causes incorrect processing of parameters of other types) and transform these templates;
- test transformed and existing templates in new environment, make changes if necessary;
- check the consistence of set of semantic properties – properties of different types and different meanings have to be defined by different names (for template parameters of non-semantic wiki resource it is insignificant);

- create the necessary semantic queries, include them to the corresponding pages and test the correctness of their execution;
- formalize the constructed structure of the IR knowledge base of the resource in the form of an ontological model.

4.5 Possibilities of explicit semantic markup

Despite the advantages of wiki templates, in some cases we propose to use sample pages that explicitly include semantic markup.

For example, sections "Reference" that partially duplicate information represented in the form of infoboxes are added to the wiki pages (Figure2). The example is taken from the website of the Ukrainian Electronic Encyclopedia of Education (page eduglos.iitta.gov.ua/index.php/Русова_Софія_Федорівна).

Русова Софія Федорівна	
	
Ім'я особи	Русова Софія Федорівна (іноз. <i>Rusova Sofia</i>)
Інші прізвища	Ліндфорс
Роки життя	18.02.1856 - 05.02.1940
Науковий ступінь	доктор наук
Вчене звання	
Напрями діяльності	літературний критик, вихователька, політична діячка, правозахисниця
Наукова школа	
Місце народження	Олешня, Городнянський повіт, Чернігівська губернія
Місце смерті	Прага, Протекторат Богемії та Моравії
Поховання	Ольшанський цвинтар
Alma mater	Фундуклеївська гімназія
Magnum opus	Український буквар
Відзнаки	
Вебсайт	

Sample code

```

[[Name::Русова]] [[First name::Софія]] [[Father name::Федорівна]] ("іноз. [[Name_e::Rusova Sofia]]") - [[Definition::видатний український педагог, громадсько-освітня діячка, письменниця, літературознавиця, теоретик і практик у галузі суспільного дошкільного виховання кінця XIX – початку XX ст., одна з організаторів жіночого руху]], [[Scientific degree::доктор наук]].
Місце народження - [[Place of birth::Олешня, Городнянський повіт, Чернігівська губернія]], ([[Day of birth::18]].[[Month of birth::02]].[[Year of birth::1856]] - [[Day of death::05]].[[Month of death::02]].[[Day of death::1940]]).

```

Русова Софія Федорівна (іноз. *Rusova Sofia*) - видатний український педагог, громадсько-освітня діячка, літературознавиця, теоретик і практик у галузі суспільного дошкільного виховання кінця XIX – початку XX ст., одна з організаторів жіночого руху, доктор наук.

Місце народження - Олешня, Городнянський повіт, Чернігівська губернія, Україна (18.02.1856 - 05.02.1940).

Місце навчання - Фундуклеївська гімназія.

Місце роботи Київ, Одеса, Кам'янець-Подільський, Україна, Прага, Чехія.

Напрями діяльності - освіта.

Найвизначніші праці - "Український буквар".

Основні нагороди - нагрудний знак Софія Русова.

Науковий напрям - дошкільна освіта.

Figure 2: Representation of information about person with use of TIO template and sample with semantic markup.

Advantages of use sample with explicit semantic markup:

- users can copy information from wiki page without markup elements (unlike information from infobox);
- content is indexed more quickly and correctly;
- editors who create wiki pages can see the markup elements and how they appear on the page, and this representation makes it easier to learn how to use such markup and its capabilities, whereas in semantic templates the markup elements are almost completely separated from the page editor;
- transition to new software versions does not cause problems with semantic markup indexing (unlike processing information from templates);
- editors and users view markup elements that can be used for semantic search (names of semantic properties that can be used in search);

- information can be represented more flexible (users can directly edit a specific sample without the need to edit the template).

Therefore, we propose to combine templates and samples for representation of IR semantics because these two solutions complement each other by their functions.

Regardless of the method of adding a semantic component to a wiki resource, the creation of various integrator pages can continue according to the needs of users based on the knowledge stored in the ontological model. This model formalizes information about the IR knowledge base and the semantics of its elements without direct contact with its developer. For example, in order to make built-in queries, it is necessary not only to know the correct names of the semantic properties of TIOS, but also to understand their meaning and possible values of these properties.

It is important to take into account that the creation of semantic properties and their use require indexing in the wiki resource database, and this action takes some time, and therefore the results of semantic queries does not show the consequences of semantization immediately, but only after their full indexing. The speed of indexing depends on the length of the task list and on the selected policy of their execution.

4.6 Approbation

We analyze semantization of wiki resources on three independent examples – the portal of the Great Ukrainian Encyclopedia of e-VUE (vue.gov.ua), the test version of the Ukrainian Electronic Encyclopedia of Education of UEEO (uee.gs4cms.com.ua) and the wiki resource of the Institute of Software Systems NASU (<http://wiki.isoftware.kiev.ua/>). All projects are based on MediaWiki and the semantic plug-in SMW, but they use different versions of this software, and system development and content semantics were performed on different methodological bases. Therefore, it can be assumed that the detected regularities are typical, if not for all, then for many wiki resources created in such a technological environment.

Conclusions

The semantization of wiki resources requires the use of distributed knowledge management methods and elements of ontological analysis for domain modeling. Selection of used methods depends on the semantization goals and the state of the IR at the time of decision about semantization. The choice of a pertinent model of IR knowledge base and its correct software implementation provide not only convenient navigation in the resource content, but also more complex retrieval and analytical functions.

The conducted analysis and practical investigations identify the following opportunities and limitations of SMW:

- SMW is focused on semantic representation of natural language content with multimedia elements, and not on transformation of all IR content to RDF;
- parameters of wiki template should be recognized as semantic properties of the corresponding wiki pages, but in practice they are not always correctly indexed in the IR database and requires additional checks;
- Generation of ontologies in RDF format is an additional option of SMW queries, not the main one, and therefore it has a rather limited functionality;
- SMW queries can define a conjunction of conditions (a set of categories of the resulting pages, check the presence of values of an arbitrary number of semantic properties and compare these values with constants);
- In the conditions of built-in queries, we can additionally use MediaWiki "magic words" to describe the current time, the current page and properties of other pages;

- Templates and regular MediaWiki pages allow certain logical operations for parser to refine search (for example, a conditional operator “if” to specify which query to execute), but more complex calculations are not supported, and results of their processing can depend on software version and settings;
- Simple (linear) conversions of semantic property values into other measurement units (miles into kilometers, kilograms into grams) are supported;
- For more complex queries (for example, with the disjunction of conditions, with complex arithmetic operations into the query conditions between the values of different properties), we need to choose another platform of Wiki semantization – for example, with SPARQL support.

Thus, the proposed review of SMW is only one component of the analysis of Wiki semantization that should be supplemented by a review with similar characteristics of other wiki platforms, such as KiWi, OntoWiki, Freebase. But such reviews should be created by specialists who use appropriate platforms to implement practical tasks.

References

- [1] J. Rogushina, A three-dimensional model of semantic search: queries, resources, and results, in: Problems in programming, (4), 2023, pp.39–55.
- [2] P. Cudré-Mauroux, Semantic Search (2019). <https://exascale.info/assets/pdf/cudre2018abigdata.pdf>.
- [3] Y. Gil, V. Ratnakar, A Comparison of (Semantic) Markup Languages, FLAIRS (2002) 413–418. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=aaa88fae632c3e19675cfe65d5f6e3730342842e>.
- [4] M. Arenas, G. Gottlob, A. Pieris, Expressive languages for querying the semantic web, in: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2014, pp.14-26.
- [5] C. Mangold, A survey and classification of semantic search approaches. International Journal of Metadata, Semantics and Ontologies, 2(1), 2007, 23-34.
- [6] P. Haase, D. Herzig, M. Musen, M., T. Tran, Semantic wiki search, in: The Semantic Web: Research and Applications: 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece, Proceedings 6, Springer Berlin Heidelberg, 2009, pp. 445-460.
- [7] Y. Koren, Working with MediaWiki, San Bernardino, CA, USA: WikiWorks Press. 2012, pp. 157–159.
- [8] M. Krötzsch, D. Vrandečić, M. Völkel, Semantic mediawiki, in: International semantic web conference. Berlin, 2006, pp. 935–942.
- [9] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, R. Studer, Semantic wikipedia, in: Proceedings of the 15th international conference on World Wide Web, 2006, pp. 585-594.
- [10] J. Rogushina, I. Grishanova, Ontological methods and tools for semantic extension of the MediaWiki technology, in: Proc. of the 12th International Scientific and Practical Conference of Programming UkrPROG, CEUR Workshoop Proceedings, 2021, Vol-2866, pp.61-73.