Collective Knowledge Engineering with Semantic Wikis

Grzegorz J. Nalepa
(AGH University of Science Technology
Kraków, Poland
gjn@agh.edu.pl)

Abstract : In the paper application of semantic wikis as knowledge engineering tool in a collaborative environment is considered. Selected aspects of semantic wikis are discussed. The main apparent limitation of existing semantic wikis is the lack of expressive knowledge representation mechanism. Building a knowledge base with a semantic wiki becomes complicated because of its collective nature, where number of users collaborate in the knowledge engineering process. A need for knowledge evaluation and analysis facilities become clear. The paper discusses a new semantic wiki architecture called *PlWiki*. The most important concept is to provide a strong knowledge representation and reasoning with Horn clauses-based representation. The idea is to use Prolog clauses on the lower level to represent facts and relations, as well as define rules on top of them. On the other hand a higher-level Semantic Web layer using RDF support is provided. This allows for compatibility with Semantic Media Wiki while offering improved representation and reasoning capabilities. Another important idea is provide an extension to already available flexible wiki solution (DokuWiki) instead of modifying existing wiki engine. Using the presented architecture it is possible to analyze rule-based knowledge stored in the wiki. [1] [2]

Key Words: semantic wikis, knowledge engineering, knowledge evaluation
Category: H.1.2, H.3, H.5

1 Introduction

The Web has become the main platform for the electronic data sharing. New technologies such as RDF allow for metadata representation, and semantic annotation. The development of the Semantic Web [Berners-Lee et al. 2001] initiative promises future knowledge sharing and engineering. In its architecture a number of higher level semantic facilities built on top of the Web would allow not just to *search data* but to *reason with knowledge*. In fact, this was the point where the focus of the Web development moved from *content* (data) to *knowledge* (in a broad sense). A number of semantic technologies is available and widely used, starting from the data structuring XML, to meta-data annotations with RDF and ontologies with RDFS and OWL. While these technologies provided knowledge encoding and representation solutions, the challenge remains to provide an efficient knowledge processing and reasoning with rules on the Web. This

[1] The paper is supported by the *BIMLOQ* Project funded from 2010–2012 resources for science as a research project.

[2] This paper is an extended version of the paper [Nalepa (2009)] presented at the ICCCI conference in Wrocław.

is in fact the point, where most of the current Semantic Web research focuses. Recent rule standards from W3C include RIF and SWRL. However, number of challenges related to effective knowledge engineering on the Web remain.

Besides knowledge representation and reasoning, a sensible knowledge engineering solution for the Web is another important challenge. One of the main aspects of the Web-based knowledge engineering is its collective nature. Large number of users provide vast amounts of data, expressed in different moments, different locations, and often in different form. New technical solutions to support this process have been provided. Social networks, that provide specific services on top of the Web and the Semantic Web, try to cope with these problems. Recently the wiki technology has gained importance with respect to the collaborative knowledge acquisition and engineering. The development of *semantic wikis* , such as IkeWiki [Schaffert et al. 2006], Semantic MediaWiki [Krötzsch et al. 2006], and SweetWiki [Buffa et al. 2008a], allowed to use the Semantic Web methods and tools on top of the existing content-centered wiki solutions. A research aiming at the analysis of social collaborations in such systems is also active [Jung and Nguyen 2008, Jung and Nguyen 2009]. Number of new solutions use the so-called Web 2.0 tools and methods to enrich knowledge acquisition and managament, e. g. see [Jung (2009), Razmerita et al. 2009].

Existing semantic wikis allow for an introduction of semantic information (e.g. meta-data, ontologies) into a wiki. In fact, they often allow to build a wiki around an ontology, which improves their conceptual coherence. Most of the semantic wikis reached a stage where reasoning capabilities have to be added. This is where some limitations of existing solutions become exposed.

The main apparent limitation of existing semantic wikis is the lack of expressive knowledge representation mechanism [van Harmelen et al. 2007]. Building a knowledge base with a semantic wiki becomes complicated because of its collective nature, where number of users collaborate in the knowledge engineering process. A need for knowledge evaluation and analysis facilities become clear. Overcoming these limitation is *the main motivation* for the research results presented in this paper.

In this paper these problems are addressed with a use of an extended semantic wiki architecture called *PlWiki* [Nalepa (2009)]. The most important concept is to provide a strong knowledge representation and reasoning with Horn clauses-based representation. But instead of building directly on top of OWL [van Harmelen and McGuinness 2004] and SWRL [Horrocks et al. 2004] a more generic solution is proposed. The idea is to use Prolog [Bratko (2000)] clauses to represent facts and define rules on the lower level. On the other hand a higher level Semantic Web layer using RDF and OWL support is provided. Another important idea is not to develop an entirely new wiki engine (e.g. [Schaffert (2006)]) but to provide an extension to an existing well-established wiki solution. It is argued, that such a generic architecture is both more flexible and efficient.

The rest of the paper is organized as follows: In [Section 2] wikis as web-based collective content engineering systems are discussed. Then in [Section 3] the development of semantic wikis is considered. The PlWiki system is discussed in [Section 4], including the requirements for the PlWiki system, the design of the PlWiki plugin for the DokuWiki. The use of the plugin is considered on an example. and the semantic layer is also discussed. PlWiki opens some new opportunities for knowledge evaluation in the wiki as discussed in [Section 5]. Finally a short evaluation of the approach as well as directions for the future work are given in [Section 6].

2 Wikis as Collective Content Engineering Solution

Wiki systems appeared in the mid 90s. According to Wikipedia the first system called "wiki" (WikiWikiWeb) was established 15 years ago. Tho goal of these systems was to provide a conceptually simple tool for massively collaborative knowledge sharing and social communication. Wikis were meant to help build certain communities interested in given topics. Clearly some of them grew large and general, such as the Wikipedia.

A wiki system is a community-driven collaboration tool. It allows users to build content in the form of the so-called wiki pages, as well as uploaded media files. Wikipages are plain text documents containing special wiki markup (e.g. for structuring content) thus creating the so-called wikitext. The wikitext is simplistic and human readable, making it a much more accessible tool than HTML/XML. Pages are identified by a unique keyword (name) and usually grouped within the so-called namespaces. Pages are linked to each other and to external websites creating a hyperwikitext structure.

An important feature of wikis is the integrated version control functionality, very helpful in a collaborative environment. It allows registering all subsequent versions of every page, thus allowing to see introduced differences. All wiki edits may be identified by user names and time stamps, so it is possible to recreate any previous state of the wiki at any given time.

From the technical point of view a wiki has a regular web-based client-server architecture. It is run on the web server and accessed by a regular browser. Wikis introduce a range of access control mechanisms from simple ones, to full-fledged ACL (Access Control Lists) solutions. On the server side wikis require different runtime environment (e.g. PHP/JSP/Python), possibly with a relational database system. A comprehensive comparison of different wiki systems can be found on `http://www.wikimatrix.org`.

One of the most interesting wiki systems for developers is DokuWiki (`http://www.dokuwiki.org`). It is designed to be portable, easy to use and set up.

Like number of other solutions DokuWiki is based on PHP. However, it does not require any relational database back-end. It allows for image embedding, and file upload and download. Pages can be arranged into namespaces which act as a tree-like hierarchy similar to directory structure. It provides syntax highlighting for in-page embedded code of programming languages such as: C/C++, Java, XML and others, using GeSHi (`qbnz.com/highlighter`). Furthermore, it supports extensive user authentication and authorization mechanisms including ACL. Its modularized architecture allows the user to extend DokuWiki with plugins which provide additional syntax and functionality. A large number of plugins is available. The templates mechanism provides an easy way to change the presentation layer of the wiki.

All wiki systems provide an abstract representation of the content they store. They all provide standard searching capabilities. However, they lack facilities helping in expressing the semantics of the stored content[3]. This is especially important in the case of collaborative systems, where number of users work together on the content. This is why wikis became one of the main applications and testing areas for the Semantic Web technologies.

3 Semantic Wikis

A step in the direction of enriching standard wikis with the semantic information has been performed by the introduction of the so-called *semantic wikis*, such as the IkeWiki [Schaffert (2006)], OntoWiki [Auer et al. 2006], SemanticMediaWiki [Krötzsch et al. 2007a], or SweetWiki [Buffa et al. 2008b]. In such systems the standard wikitext is extended with semantic annotations. These include relations (represented as RDF triples) and categories (here RDFS is needed). It is possible to query the semantic knowledge with a simple quer ylanguage or SPARQL [Seaborne and Prud'hommeaux 2008] thus providing dynamic wiki pages. Ultimately these extension can also allow for building an ontology of the domain to which the content of the wiki is related. This extension introduces not just new content engineering possibilities, but also semantic search and analysis of the content.

Number of semantic wiki systems are available, most of them in the development stage providing demo versions[4]. In a recent paper [Buffa et al. 2008b] a comprehensive overview of semantic wikis technology has been given, with number of systems described w.r.t to their features and implementation details. The development of semantic wikis is very dynamic. Since that paper has been written some of these systems are no longer supported, e.g. IkeWiki. Some important features of selected systems available up to recently are presented in the

[3] Besides simple tagging mechanisms, that can later be used to create the so-called *folksonomies.*

[4] See `http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art`.

[Tab. 1]. A recent FP7 project Kiwi (`http://www.kiwi-project.eu`) aims at providing a collaborative knowledge management based on semantic wikis (it is the continuation of IkeWiki effort).

Feature	*AceWiki*	*IkeWiki*	*Kiwi*	*KnowWE*	*SMW*	*SweetWiki*
Active	+	–	+	+	+	–
License	LGPL	GPL	CDDL	LGPL	GPL	LGPL
Platform	Java	Java	Java	Java	PHP	Java
Reuses engine	no	no	no	yes	yes	no
Wiki object model	no	no	yes	no	no	yes
Assisted annotations	yes	yes	yes	yes	no	yes
Social tagging	no	yes	yes	yes	no	yes
Edit ontology	yes	yes	yes	yes	yes	no
Represen.	ACE	OWL	OWL	OWL	RDFS	RDFS
Queries	no	SPARQL	SPARQL	SPARQL	wikiml	SPARQL
Reasoning	no	Jena	misc	d3web	KAON	Corese
Wikipages Versioning	no	yes	yes	yes	yes	yes
Metadata Versioning	no	no	partial	partial	partial	partial
Discussion pages	no	yes	yes	yes	no	no

Table 1: Selected semantic wikis features

From the knowledge engineering point of view expressing basic semantics is not enough. In fact a knowledge-based system should provide effective knowledge representation and processing methods. In order to extend semantic wikis to knowledge-based systems, ideas to use a problem-solving knowledge have been introduced. An example of such a system is the *KnowWE* semantic wiki [Baumeister et al. 2007, Baumeister (2008), Reutelshoefer et al. 2008]. In such a system the semantic knowledge is extended with the problem-solving domain-specific knowledge. The system allows for introducing knowledge expressed with decision rules and trees related to the domain ontology. It could be said, that conceptually it is built on top of the simpler wikis, e.g. the SMW.

In this paper a generic solution based on the use of Prolog as the language for expressing both the knowledge semantics, and processing is presented.

4 PlWiki System Overview

4.1 Motivation and Objectives

The semantic wiki technology is a young one. Multiple systems are developed to test new ideas and features. However, number of conceptual challenges remain. Some of the persistent problems are: an expressive yet effective (in terms of inference) knowledge representation, allowing for explicitly representing the semantics of the wiki content, powerful query and inference facilities, that enable reasoning on top of the gathered knowledge; in fact they are the main factor limiting the practical usability of the knowledge, interfaces helping users to encode and use the knowledge they poses, and integration with the existing technologies that improves the portability and makes the development of the system easier.

All of the existing semantic wiki solutions address these problems in various manners [Buffa et al. 2008b]. Some of the most common approaches to cope with these problems include the extensive use of selected Semantic Web technologies to introduce well-founded semantics. This includes the use of RDF for meta-data, as well as RDFS, and possibly OWL, for ontology management, and SPARQL as the query language [Krötzsch et al. 2007a].
Some other [Reutelshoefer et al. 2008] introduce extended knowledge representation for problem solving. User interfaces are usually based on simple forms helping to input semantic annotations, as well as editors highlighting the wiki markup [Buffa et al. 2008b]. Most of these solutions modify some existing wiki engines, e.g. Media Wiki that powers Wikipedia. In general, they still lack universal rule representation (mostly due to the development stage of SWRL). However, it is an ongoing research where an optimal solution is hard to find.

The approach discussed in this paper is different. The basic idea is to allow the use of a logical knowledge representation based on Horn clauses [Ben-Ari (2001)] for facts, relations and rules, as well as dynamic queries. This allows not only to represent facts, but also introduce rules for inference. On top of this the Semantic Web layer with RDF may also be provided. The approach is based on the concept of using the Prolog language interface. This also opens up possibilities of powerful querying mechanism, more powerful than SPARQL (while compatibility layer compatible with SPARQL is provided). The solution is developed as an optional extension to an existing modular wiki engine of DokuWiki.

The main objectives of this approach are to enhance both representation and inference features, allow for a complete rule framework in the wiki, and to use a clean integration approach with an existing system.

A decision has been made to build the new wiki with use of the Prolog language [Bratko (2000)]. This allows to provide rich knowledge representation, including rules, as well as allow for an efficient and flexible reasoning in the wiki, and expressive power equivalent to Horn clauses. Thus it is possible the represent the domain specific knowledge and reasoning procedures with the same generic representation.

The system provides a Semantic Web layer for the knowledge engineer. This means the support for meta-data encoding with RDF and ontologies with RDFS (possibly with OWL too). Such a solution allows to extend the wiki using existing ontologies, optionally build with other semantic wikis.

Based on number of experiences with other semantic wiki engines. a decision has been made to extend an existing well-established wiki engine, There are tens of wiki engines available (see `www.wikimatrix.org`). Most of them are similar w.r.t to main concepts and features. However, there are number of differences when it comes to the wikitext syntax, implementation and runtime environment, as well as extra features. This is why, instead of building yet another wiki engine, or modify an existing one, another solution is proposed. The idea is to use a ready, flexible and extensible wiki engine, that could be optionally extended with knowledge representation and processing capabilities.

Considering the collective knowledge engineering environment it is important to allow for meta-knowledge representation suitable for wiki knowledge evaluation. PlWiki has the ability to analyze wiki knowledge using procedures specified in the wiki, using the same representation as the wiki contents.

In order to meet these requirement a fast, flexible and portable Prolog implementation has been chosen. It provides a rich library stack for the Semantic Web compatibility. In the next section the design of the PlWiki extension for the DokuWiki system built with use of the SWI-Prolog environment is given.

4.2 System Design and Architecture

The main goal of the new knowledge wiki design is to deliver a generic and flexible solution. Instead of modifying an existing wiki engine or implementing a new one, a development of an extension of the DokuWiki system was chosen. To provide a rich knowledge representation and reasoning for the Semantic Web, the SWI-Prolog environment was selected. The basic idea is to build a layered knowledge wiki architecture, where the expressive Prolog representation is used on the lowest knowledge level. This representation is embedded within the wiki text as an optional extension. On top of it number of layers are provided. These include standard meta-data descriptions with RDF and ontologies specification solutions with RDFS and OWL.

The PlWiki stack can be observed in [Fig. 1]. The stack is based on a simple runtime including the Unix environment with the Unix filesystem, the Apache

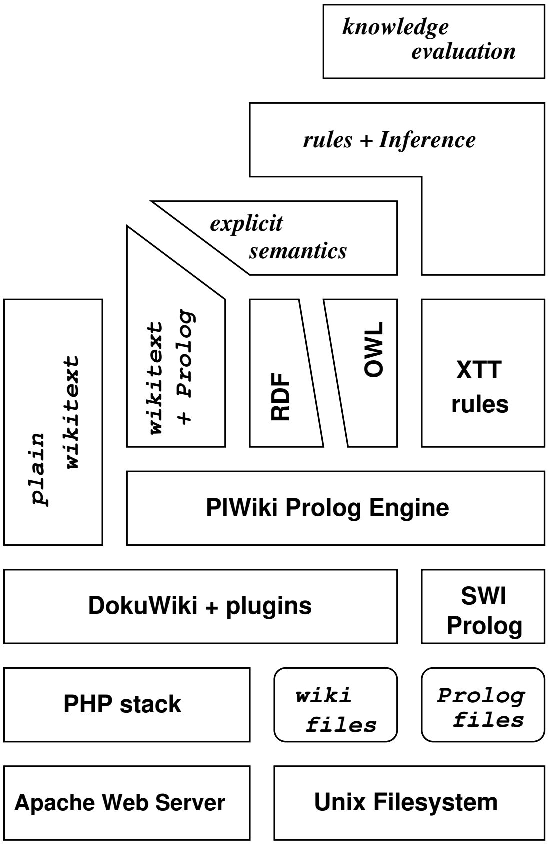


Figure 1: PlWiki architecture

web server and the PHP stack. Using this runtime the standard DokuWiki installation is run. The PlWiki functionality is implemented with the use of an optional plugin allowing to enrich the wikitext with Prolog clauses, as well run the SWI-Prolog interpreter. It is also possible to extend the wikitext with explicit semantical information encoded with the use of RDF and possibly OWL representation. This layer uses the Semantic Web library provided by SWI-Prolog. An optional decision rule layer is also considered with the use of the HeaRT runtime for the XTT^2 framework [Nalepa and Ligęza 2005, Nalepa and Ligęza 2008].

The main layer interfacing with the DokuWiki engine is presented in [Fig. 2]. The figure shows the dataflow in the DokuWiki system. DokuWiki provides a flexible plugin system, providing five kinds of plugins (see `www.dokuwiki.org/devel:plugins`):

- *Syntax Plugins*, extending the wikitext syntax,

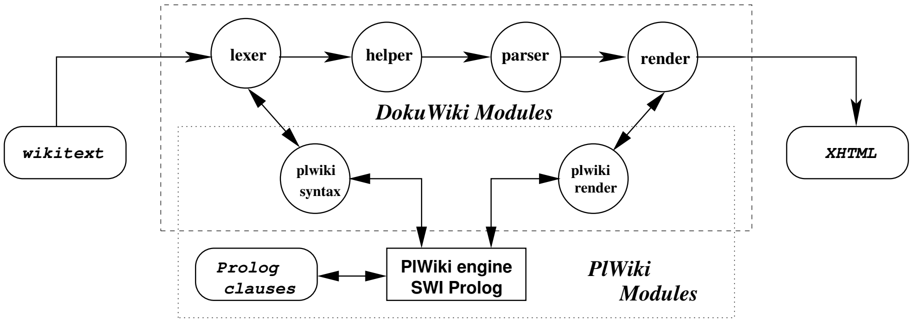


Figure 2: PlWiki modules

- *Action Plugins*, redefining selected core wiki operations, (e.g. saving pages),
- *Admin Plugins*, providing extra administration functionality,
- *Helper Plugins*, supporting other plugins with generic functions,
- *Renderer Plugins*, allowing to create new export modes (possibly replacing the standard XHTML renderer).

The current version of PlWiki implements both the *Syntax* and *Renderer* functionality. Text-based wikipages are fed to a lexical analyzer (Lexer) which identifies the special wiki markup. The standard DokuWiki markup is extended by a special `<pl>...</pl>` markup that contains Prolog clauses. The stream of tokens is then passed to the Helper that transforms it to special renderer instructions that are parsed by the Parser. The final stage is the Renderer, responsible for creating a client-visible output (e.g. XHTML). In this stage the second part of the plugin is used for running the Prolog interpreter.

The detailed functionality of the PlWiki Syntax Plugin includes parsing the Prolog code embedded in the wikitext, and generating the knowledge base composed of files containing the Prolog code, where each wikipage has a corresponding file in the knowledge base. The PlWiki Renderer plugin is responsible for executing the Prolog interpreter with a given goal, and rendering the results via the standard DokuWiki mechanism.

The PlWiki framework uses the SWI-Prolog environment, licensed under the Lesser GNU Public License (see `www.swi-prolog.org`). It is a mature implementation widely used in research and education as well as for commercial applications. It provides a fast and scalable development environment, including graphics, libraries and interface packages, portable to many platforms, including Unix/Linux platforms, Windows, and MacOS X. SWI-Prolog provides a rich set of libraries, including the *semweb* library for dealing with standards from the

W3C standard for the Semantic Web (RDF, RDFS and OWL). This infrastructure is modular, consisting of Prolog packages for reading, querying and storing Semantic Web documents.

One should keep in mind, that the Prolog-based representation is quite close to the natural language. Not only on the semantical level, but to a degree also on the syntactic level. It is possible thanks to the operator redefinition.

Currently the PlWiki system is under heavy development. The prototype of the system was developed. Next versions are currently being developed. See `http://home.agh.edu.pl/gjn` for more information.

4.3 Knowledge Representation Features

Below basic use examples of the generic Prolog representation are given.

```
<pl> capital(germany,berlin). country(germany). country(poland). </pl>
```

This simple statement adds two facts to the knowledge base. The plugin invocation is performed using the predefined syntax. To actually specify the goal (query) for the interpreter the following syntax is used:

```
<pl goal="country(X),write(X),nl,fail"></pl>
```

It is possible to specify a given *scope* of the query (in terms of namespaces):

```
<pl goal="country(X),write(X),nl,fail" scope="prolog:examples"></pl>
```

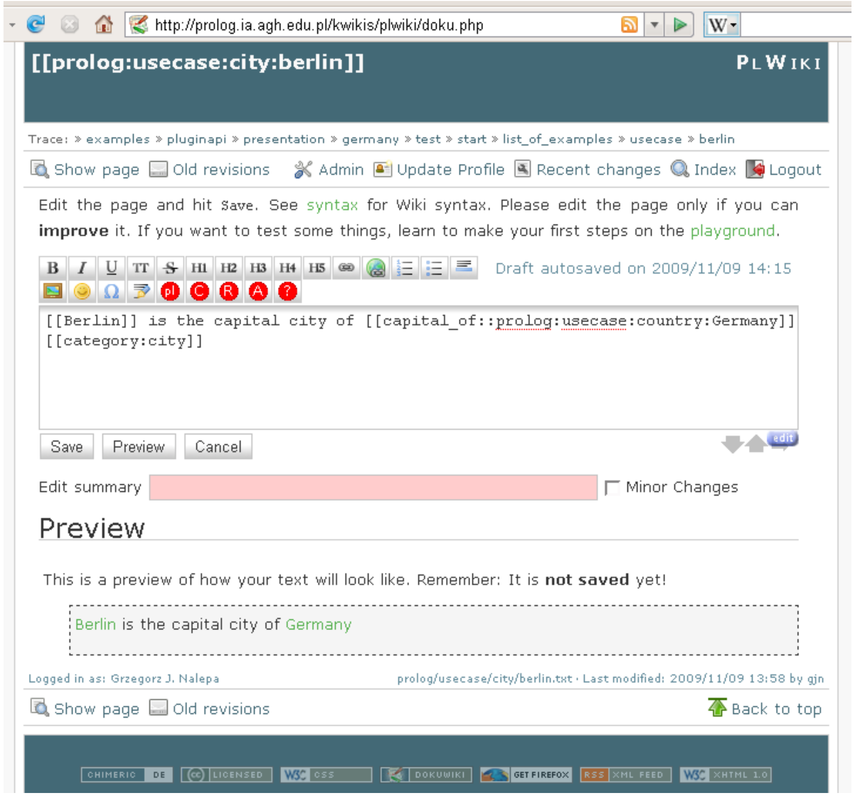
A bidirectional interface, allowing to query the wiki contents from the Prolog code is also available, e.g.:

```
<pl goal="consult('lib/plugins/prolog/plwiki.pl'),
          wikiconsult('plwiki/pluginapi'),list."></pl>
```

There are several options how to analyze the wiki knowledge base (that is Prolog files built and extracted from wiki pages). A basic approach is to combine all clauses. More advanced uses allow to select pages (e.g. given namespace) that are to be analyzed. On top of the basic Prolog syntax, semantic enhancements are possible. These can be easily mapped to Prolog clauses. An example of editing session with PlWiki can be observed in [Fig. 3].

4.4 Semantic Representation Layer

Besides the generic Prolog-based knowledge representation features based on pure Prolog clauses, typical semantic wiki features are supported. Semantic Media Wiki (SMW) [Krötzsch et al. 2007b)], a standard semantic wiki solution, provides a simple yet flexible mechanism for annotating categories, and properties. n the first version of PlWiki three main features are considered:



[[prolog:usecase:city:berlin]]

Trace: » examples » pluginapi » presentation » germany » test » start » list_of_examples » usecase » berlin

Edit the page and hit Save. See syntax for Wiki syntax. Please edit the page only if you can **improve** it. If you want to test some things, learn to make your first steps on the playground.

Draft autosaved on 2009/11/09 14:15

```
[[Berlin]] is the capital city of [[capital_of::prolog:usecase:country:Germany]]
[[category:city]]
```

Preview

This is a preview of how your text will look like. Remember: It is **not saved** yet!

Berlin is the capital city of Germany

Logged in as: Grzegorz J. Nalepa

prolog/usecase/city/berlin.txt · Last modified: 2009/11/09 13:58 by gjn

Figure 3: PlWiki editing session

- categories definitions as in SMW,
- simple queries from SMW (with SPARQL queries in the future), and
- generic RDF annotations.

To provide a better compatibility with existing solutions parsing of SMW wikitext is provided, with a corresponding Prolog representation available. Wiki user can use the SMW syntax directly in DokuWiki to enter wikitext. The PlWiki plugins transforms the wikitext to Prolog clauses, asserted to the internal knowledge base. In fact these clauses could also be introduced by using the PlWiki `<pl></pl>` tags. Examples are shown below, with the SMW syntax given first, and the corresponding Prolog representation after it.

```
Berlin is the capital city of [[capital_of::Germany]] [[category:city]]
  wiki_category('City','Berlin').
  wiki_property(capital_of,subject_page_name,'Germany').
Germany is a country in central Europe.
```

```
[[category:country]] [[location:=Central Europe]]
  wiki_category('Country,'Germany').
  wiki_attribut(page_uri,location,'Central_Europe').
```

The Prolog clauses are asserted to the PlWiki knowledge base by the syntax plugin analyzing the wiki text. In a similar fashion simple queries are handled. A query for a category or property is simply mapped to a Prolog goal:

```
{{#ask: [[category:city]] [[capital of::Germany}}
  wiki_category('Cities',Page),
  wiki_property(capital_of,Page,'Germany'), wiki_out(Page).
```

Plain RDF annotations are also supported. Currently, these are separated from the explicit annotations mentioned above. For compatibility reasons an RDF annotation can be embedded directly in XML serialization, then it is parsed by the corresponding Prolog library, and turned to the internal representation, that can also be used directly. SWI-Prolog's represents RDF triples simply as: `rdf(?Subject, ?Predicate, ?Object).` So mapping the above example would result in: `rdf('Berlin',capital_of,'Germany').` The SWI-Prolog RDF storage in highly optimized. It can be integrated with the provided RDFS and OWL layers, as well as with the *ClioPatria* platform[5] providing also SPARQL queries.

Thanks to the full Prolog engine available in the wiki, the inference options are almost unlimited. Prolog uses backwards chaining with program clauses. However, it is very easy to implement meta-interpreters for forward chaining.

Using wiki knowledge it is possible to define rules, e.g.: "Nordic country is a country with `location` set to `Northern Europe`" is in Prolog:

```
<pl cache="true"> nordic_country(X) :-
                    wiki_category(X,'country'),
                    wiki_attribute(X,'location','Northern Europe'). </pl>
```

Compound queries can also be created and executed as Prolog predicates.

4.5 Applications of PlWiki

PlWiki is in an experimental development phase. Current applications include special knowledge engineering tasks, including basic rule-based reasoning tasks in the wiki, and teaching knowledge engineering classes. Future applications are planned, including dedicated knowledge intensive closed community portals. System development will focus on flexible user interfaces supporting complex knowledge representation features.

[5] See http://e-culture.multimedian.nl/software/ClioPatria.shtml.

5 Knowledge Evaluation in a Collective Environment

Semantic wikis with both fact and rule representation can be considered knowledge-based systems. The distributed knowledge development process in a wiki poses new problems in knowledge engineering when compared to the classic development of monolithic knowledge bases. In case of most of the semantic wikis the focus of the current research is on the knowledge representation, integration and authoring. However, with the growing amount of knowledge contained in semantic wikis, the knowledge quality issues seem to be critical [Baumeister and Nalepa 2009].

Formal verification of knowledge-based systems is a mature field, where a number of important results have been brought up in the last decades. For some important contributions see [Preece (1992), Preece (1993)] and [Lunardhi and Passino 1995, Vermesan (1997)]. The research in verification has been especially active in the field of rule-based expert systems [Giarratano and Riley 2005]. A taxonomy of formal properties for the verification of such systems has been presented in [Vermesan and Coenen 1999], with some more recent follow-ups such as [Ligęza (2006)].

According to [Vermesan and Coenen 1999, Ligęza (2006)] three most important groups of properties in the verification of knowledge-based systems include the following: *Consistency* of the knowledge base means that no contradictory conclusions can be inferred from valid facts. *Completeness* of knowledge means – in a vague sense – that no information is missing. *Conciseness* means, that no redundant (unnecessary) knowledge can be found in the knowledge base. Let us consider how the three classic rule verification criteria may be applied with respect to the distributed knowledge base in a wiki system.

Wikis are composed of wiki *pages*. So a wiki can be described as a distributed knowledge-based system, where a number of knowledge bases exist. It is a distributed system, because everyone can work on his own knowledge base. Pages are usually grouped within *namespaces* related to their common semantics, which can be explicitly marked in a semantic wiki ontology. Pages in different namespaces can be interconnected, as well as pages can reference pages in other wikis (interwiki connections).

Considering the general knowledge wiki architecture several *verification scopes* need to be considered:

- single page scope – where the given property is analyzed only in a single wiki page, and all links are ignored.

- namespace scope – where every page in a group is considered to be a component of a single namespace-wide knowledge base, so the given formal property must hold with respect to the whole group. This means that all the links to the pages in the namespace have to be considered, whereas external links are ignored.

- wiki scope – this global wiki scope treats the whole wiki as a single knowledge base, interwiki wiki links are ignored.
- interwiki scope – in this most complex case interwiki links should be analyzed. In this paper this context is ignored, simply because current technical solutions and lack of standards make this case almost impossible to consider practically.

With respect to the above scopes the properties may be interpreted as follows.

Wiki Consistency means that no contradictory information is contained in the unit. For the practical verification the given wiki unit (page, namespace) needs to be analyzed to detect contradictory *facts* or *rules*. Inconsistency is likely to appear in a distributed environment such as wiki, where a number of independent authors extend the knowledge base. It should be detected on-line, during the wiki editing session. However, it is worth noting, that considering the evolutionary nature of the wiki knowledge, such an inconsistency between two versions of a given page could be in fact a hint for knowledge refinement, so it is not obvious which of the above contradictory facts is “correct” considering the changed page contents.

Wiki Completeness means that no information with respect to the given ontology is missing in the unit. This should be considered with respect to all the pages and knowledge bases in the wiki. In a general case completeness verification is hard. It is possible to conduct such a procedure in cases where the domain of a certain property is given. This is possible in case of wikis designed according to a domain ontology.

Wiki Conciseness can be interpreted as a state where no redundant information is contained in the unit. In general cases it means that no identical facts or rules are inserted. In more specific cases it could also mean that new facts are more general, or that new rules subsume the older versions. In case of multi-page scope (namespace,wiki) practical implementations of the verification algorithms should consider comparing pairs of pages as units for properties verification.

Using the Prolog interpreter selected evaluation plugins are provided as a part of Plwiki. Thus it is possible to analyze the knowledge stored in the wiki. This is an experimental feature that is being developed. It is presented in [Fig. 1] as the top layer.

6 Future Development

Considering the current prototype of the system, number of directions for future work are evaluated. In the current version of the prototype it is not possible to check the syntax of the Prolog code, or assist the user in entering it. In the future debugging and syntax highlighting features are planned.

There are obviously some performance issues regarding knowledge processing within a wiki system based on Prolog code interpretation. Extracting knowledge from many pages and processing it could be a time consuming operation. Some smart caching techniques are evaluated. They are based on the caching mechanism present in the DokuWiki system.

An enhanced direct support for wiki markup present in other solutions (e.g. SemanticMediaWiki) is also planned. This should ultimately allow to import ready to use semantic wikis implemented with SMW, and possibly other wikis.

The user interface is an important area for improvement. Besides Prolog editing support, extended interfaces that use semantic forms are also considered. They should also allow ontology edition and visualization.

An important challenge is the rule framework in the wiki. Thanks to Prolog flexibility, it is possible to support number of rule languages built on top the SWI stack, including Semantic Web languages such as SWRL (with Description Logics reasoners integrated), as well as custom solutions. One of the options that is evaluated is the use of XTT^2 rule framework [Nalepa and Ligęza 2008] that has a transparent Prolog implementation.[6] On the other hand it provides standalone visual knowledge editors, and formal analysis tools (e.g. for rule verification).

The main challange for the future development of the wiki is a complete redesign of the system based on the proposed *Loki* architecture. The principal idea in *Loki* (*Lo*gical *Wi*ki) is to allow for rich knowledge representation features in the wiki including: semantic metadata annotations (e.g. RDF, or SMW mechanisms), taxonomies with RDFS, ontologies with OWL, decision tables and decision rules with XTT2. In Loki all of these are internally represented using Prolog constructs. Then the builtin Prolog interpreter processes the knowledge allowing for reasoning.

7 Concluding Remarks

In the paper application of semantic wikis as knowledge engineering tool in a collaborative environment is considered. Selected aspects of semantic wikis are discussed. The main apparent limitation of existing semantic wikis is the lack of expressive knowledge representation mechanism. Building a knowledge base with a semantic wiki becomes complicated because of its collective nature, where number of users collaborate in the knowledge engineering process. A need for knowledge evaluation and analysis facilities become clear.

The paper discusses an idea of implementing a semantic wiki using Prolog for knowledge representation and inference. The system allows for fact, relations

[6] See HeKatE project website at `http://hekate.ia.agh.edu.pl`.

and rule representation, using Prolog. It also allows to use Semantic Web languages such as RDF to provide semantic annotations, opening possibilities to implement full support for OWL-based ontologies. The distributed and collaborative knowledge development process in a wiki poses new problems in knowledge engineering when compared to the classic development of monolithic knowledge bases. With the growing amount of knowledge contained in semantic wikis, the knowledge quality issues seem to be critical. In PlWIki several knowledge analysis plugins are provided. Thus it is possible to analyze the knowledge stored in the wiki w.r.t. to a number of features [Baumeister and Nalepa 2009].

References

[Auer et al. 2006] Auer, S., Dietzold, S., Riechert, T.: “Ontowiki - a tool for social, semantic collaboration”; I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, L. Aroyo, eds., International Semantic Web Conference; volume 4273 of Lecture Notes in Computer Science; Springer, (2006), 736–749.

[Baumeister (2008)] Baumeister, F., J.; Puppe: “Web-based knowledge engineering using knowledge wikis”; Proc. of the AAAI 2008 Spring Symposium on ”Symbiotic Relationships between Semantic Web and Knowledge Engineering”; Stanford University, USA,(2008), 1–13.

[Baumeister and Nalepa 2009] Baumeister, J., Nalepa, G. J.: “Verification of distributed knowledge in semantic knowledge wikis”; H. C. Lane, H. W. Guesgen, eds., FLAIRS-22: Proceedings of the twenty-second international Florida Artificial Intelligence Research Society conference: 19–21 May 2009, Sanibel Island, Florida, USA; FLAIRS; AAAI Press, Menlo Park, California, (2009), 384–389.

[Baumeister et al. 2007] Baumeister, J., Reutelshoefer, J., Puppe, F.: “KnowWE: community-based knowledge capture with knowledge wikis”; K-CAP ’07: Proceedings of the 4th international conference on Knowledge capture; ACM, New York, NY, USA, (2007), 189–190.

[Ben-Ari (2001)] Ben-Ari, M.: Mathematical Logic for Computer Science; Springer-Verlag, London, (2001).

[Berners-Lee et al. 2001] Berners-Lee, T., Hendler, J., Lassila, O.: “The semantic web: Scientific american”; Scientific American; (2001).

[Bratko (2000)] Bratko, I.: Prolog Programming for Artificial Intelligence; Addison Wesley, (2000), 3rd edition.

[Buffa et al. 2008a] Buffa, M., Gandon, F., Ereteo, G., Sander, P., Faron, C.: “SweetWiki: A semantic wiki”; Web Semantics; 8 (2008), 1, 84–97.

[Buffa et al. 2008b] Buffa, M., Gandon, F., Ereteo, G., Sander, P., Faron, C.: “SweetWiki: A semantic wiki”; Web Semantics: Science, Services and Agents on the World Wide Web; (2008), in press.

[Giarratano and Riley 2005] Giarratano, J. C., Riley, G. D.: Expert Systems; Thomson, (2005).

[Horrocks et al. 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M.: “SWRL: A semantic web rule language combining OWL and RuleML, w3c member submission 21 may 2004”; Technical report; W3C (2004).

[Jung (2009)] Jung, J. J.: “Knowledge distribution via shared context between blog-based knowledge management systems: A case study of collaborative tagging”; Expert Syst. Appl.; 36 (2009), 7, 10627–10633.

[Jung and Nguyen 2008] Jung, J. J., Nguyen, N. T.: “Collective intelligence for semantic and knowledge grid”; J. UCS; 14 (2008), 7, 1016–1019.

[Jung and Nguyen 2009] Jung, J. J., Nguyen, N. T.: “Consensus choice for reconciling social collaborations on semantic wikis”; N. T. Nguyen, R. Kowalczyk, S.-M. Chen, eds., ICCCI; volume 5796 of Lecture Notes in Computer Science; Springer, 2009, (472–480).

[Krötzsch et al. 2006] Krötzsch, M., Vrandecić, D., Völkel, M.: “Semantic MediaWiki”; ISWC’06: Proceedings of the 5th International Semantic Web Conference, LNAI 4273; Springer, Berlin, (2006), 935–942.

[Krötzsch et al. 2007a] Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: “Semantic wikipedia”; Web Semantics; 5 (2007), 251–261.

[Krötzsch et al. 2007b)] Krötzsch, M., Vrandecić, D., Völkel, M., Haller, H., Studer, R.: “Semantic Wikipedia”; Web Semantics; 5 (2007), 4, 251–261.

[Ligęza (2006)] Ligęza, A.: Logical Foundations for Rule-Based Systems; Springer-Verlag, Berlin, Heidelberg, (2006).

[Lunardhi and Passino 1995] Lunardhi, A. D., Passino, K. M.: “Verification of qualitative properties of rule-based expert systems”; Applied Artificial Intelligence; 9 (1995), 587–621.

[Nalepa (2009)] Nalepa, G. J.: “Plwiki - a generic semantic wiki architecture”; N. T. Nguyen, R. Kowalczyk, S.-M. Chen, eds., Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, First International Conference, ICCCI 2009, Wroclaw, Poland, October 5-7, 2009. Proceedings; volume 5796 of Lecture Notes in Computer Science; Springer, (2009), 345–356.

[Nalepa and Ligęza 2005] Nalepa, G. J., Ligęza, A.: “A graphical tabular model for rule-based logic programming and verification”; Systems Science; 31 (2005), 2, 89–95.

[Nalepa and Ligęza 2008] Nalepa, G. J., Ligęza, A.: “XTT+ rule design using the alsv(fd)”; A. Giurca, A. Analyti, G. Wagner, eds., ECAI 2008: 18th European Conference on Artificial Intelligence: 2nd East European Workshop on Rule-based applications, RuleApps2008: Patras, 22 July 2008; 11–15; University of Patras, Patras, (2008).

[Preece (1992)] Preece, A. D.: “Verification, validation, and test of knowledge-based systems”; AI Magazine; 13 (1992), 4, 77.

[Preece (1993)] Preece, A. D.: “A new approach to detecting missing knowledge in expert system rule bases”; Int. J. of Man-Machine Studies; 38 (1993), 161–181.

[Razmerita et al. 2009] Razmerita, L., Kirchner, K., F, S.: “Personal knowledge management. the role of web 2.0 tools for managing knowledge at individual and organisational levels”; ONLINE INFORMATION REVIEW; 33 (2009), 6, 1021–1039.

[Reutelshoefer et al. 2008] Reutelshoefer, J., Baumeister, J., Puppe, F.: “Ad-hoc knowledge engineering with semantic knowledge wikis”; Proc. of SemWiki 2008 - The Wiki Way of Semantics, Workshop co-located with the 5th European Semantic Web Conference; Tenerife, Spain, (2008).

[Schaffert (2006)] Schaffert, S.: “Ikewiki: A semantic wiki for collaborative knowledge management”; WETICE ’06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises; IEEE Computer Society, Washington, DC, USA, (2006), 388–396.

[Schaffert et al. 2006] Schaffert, S., Gruber, A., Westenthaler, R.: “A semantic wiki for collaborative knowledge formation”; Proceedings of SEMANTICS 2005 Conference; Trauner Verlag, (2006).

[Seaborne and Prud’hommeaux 2008] Seaborne, A., Prud’hommeaux, E.: “SPARQL query language for RDF”; W3C recommendation; W3C (2008); http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[van Harmelen et al. 2007] van Harmelen, F., Lifschitz, V., Porter, B., eds.: Handbook of Knowledge Representation; Elsevier Science, (2007).

[van Harmelen and McGuinness 2004] van Harmelen, F., McGuinness, D. L.: “OWL web ontology language overview”; W3C recommendation; W3C (2004); http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[Vermesan (1997)] Vermesan, A.: The Handbook of Applied Expert Systems; chapter Foundation and Application of Expert System Verification and Validation.; CRC Press, (1997).

[Vermesan and Coenen 1999] Vermesan, A. I., Coenen, F., eds.: Validation and Verification of Knowledge Based Systems. Theory, Tools and Practice; Kluwer Academic Publisher, Boston, (1999).