Loki – Presentation of Logic-based Semantic Wiki

Weronika T. Adrian, Grzegorz J. Nalepa

AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{wta,gjn}@agh.edu.pl

Abstract. TOOL PRESENTATION: The paper presents a semantic wiki, called Loki, with strong logical knowledge representation using rules. The system uses a coherent logic-based representation for semantic annotations of the content and implementing reasoning procedures. The representation uses the logic programming paradigm and the Prolog programming language. The proposed architecture allows for rule-based reasoning in the wiki. It also provides a compatibility layer with the popular Semantic MediaWiki platform, directly parsing its annotations.

1 Motivation

Semantic wikis enrich standard wikis with the semantic information expressed by a number of mechanisms. Three basic questions every semantic wiki needs to address are[1]: 1) how to annotate content?, 2) how to formally represent content?, 3) how to navigate content? In last several years multiple implementations of semantic wikis have been developed, including IkeWiki[2], OntoWiki[3], Semantic MediaWiki[4], and AceWiki[5]. The summary of semantic wiki systems is available[6].

From the knowledge engineering point of view, simply expressing semantics is not enough. A knowledge-based system should provide both effective knowledge representation and processing methods. In order to extend semantic wikis to knowledge-based systems, ideas to use rule-based reasoning and problem-solving knowledge have been introduced. An example of such a system is the KnowWE semantic wiki[7].The system allows for introducing knowledge expressed with decision rules and trees related to the domain ontology.

Logic-based Wiki [5], or *Loki* for short, uses the logic programming paradigm to represent knowledge in the wiki. The main design principles are as follows: 1) provide an expressive underlying logical representation for semantic annotations and rules, 2) allow for strong reasoning support in the wiki, 3) preserve backward compatibility with existing wikis, namely Semantic MediaWiki.

[1] See `http://aran.library.nuigalway.ie/xmlui/handle/10379/574`.

[2] See `http://ikewiki.salzburgresearch.at/`.

[3] See `http://ontowiki.net/Projects/OntoWiki`.

[4] See `http://semantic-mediawiki.org/wiki/Semantic_MediaWiki`.

[5] See `http://attempto.ifi.uzh.ch/acewiki/`.

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

[7] See `www.is.informatik.uni-wuerzburg.de/en/research/applications/knowwe`.

2 Architecture

A prototype implementation of Loki, called PlWiki (Prolog Wiki), has been developed [2,4]. The main goal of the system design is to deliver a generic and flexible solution. Thus, instead of modifying an existing wiki engine or implementing a new one, an extension of the DokuWiki[8] system has been developed.

The architecture can be observed in Fig. 1. The stack is based on a simple runtime including the Unix environment with the Unix filesystem, Apache web server and PHP stack. Using this runtime, a standard DokuWiki installation is run. The Loki functionality is implemented with the use of Dokuwiki plugins allowing to enrich the wikitext with semantic annotations and Prolog clauses, as well as run the SWI-Prolog interpreter. It is also possible to extend the wikitext with explicit semantic information encoded using RDF and OWL representation. This layer uses the Semantic Web library provided by SWI-Prolog.

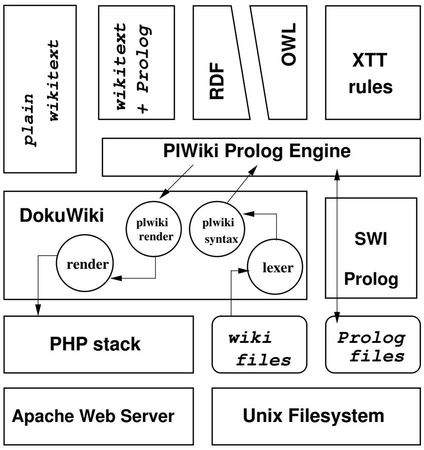


Fig. 1. Loki Architecture – PlWiki Implementation

3 Features

The main idea of Loki consists in representing the semantic annotations in a formalized way, and simultaneously enriching them with an expressive rule-based knowledge representation. Both semantic annotations and Prolog clauses (facts and rules) may be embedded within in the wiki content. The resulting knowledge base is homogeneous from the logical point of view.

SMW Support There are three main methods of semantic annotations in Semantic MediaWiki (SMW) that are supported in Loki: categories, relations and attributes. Loki also supports the query language used in SMW.

[8] See `http://www.dokuwiki.org`.

Categories, relations, attributes and queries are represented by appropriate Prolog terms. Technically, SMW markup is converted to Prolog code and then saved in a Loki file related to a Wiki page. Examples (annotations written on a page for "The Call of Cthulhu" book) are as follows, with the SMW syntax given first, and the Prolog representation below.

```
[[category:book]]
    wiki_category('book','the_call_of_cthulhu').
**Author**: [[author::h_p_lovecraft]]
    wiki_relation('the_call_of_cthulhu','author','h_p_lovecraft').
**Date**: [[date:=2011]]
    wiki_attribute('the_call_of_cthulhu','date','2011').
{{#ask: [[category:book]] [[author::h_p_lovecraft]]}}
    wiki_category('book',Page),
    wiki_relation(Page,'author','h_p_lovecraft').
```

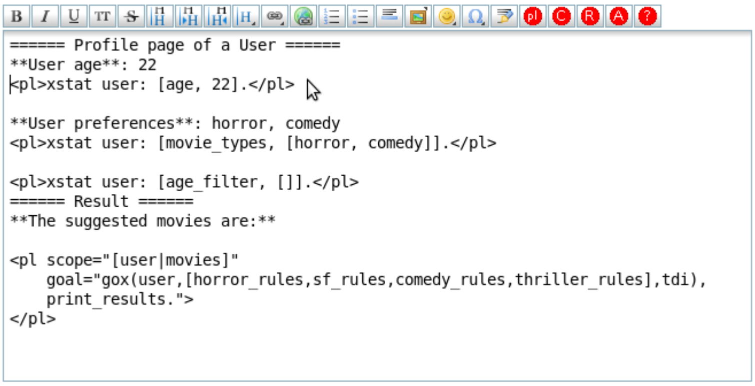
Semantic Web Standards Support RDF annotations can be embedded directly in the XML serialization. They are parsed by the SWI-Prolog *semweb/rdf_db* library, and turned to the internal representation. Within the wikitext, a SPARQL query (SELECT, ASK or DESCRIBE) may be posed. The query scope is the whole wiki system. Analogously to the SMW-like queries, SPARQL ones are also translated to Prolog goals and then executed by the wiki engine.

Semantic information gathered in Loki may be exported to the RDF/XML format. The exported file consists of a header with used namespaces, metadata of the exported page, and optionally ontological information about categories, relations and attributes. Categories are exported as OWL Classes, relations between pages as Object Properties, and attributes as Datatype Properties. Information about subcategories and subproperties is exported with the the use of `rdfs:subClassOf` and `rdfs:subProperty`.

Rule-based Reasoning An optional rule layer is provided using the HeaRT [3] runtime for the XTT2 framework [6]. XTT2 (eXtended Tabular Trees v2) knowledge representation uses attributive table format. Rules based on the same attributes are grouped within tables, and the system is split into a network of such tables representing the inference flow. XTT2 rules can be serialized into a HMR (HeKatE Meta Representation) format, supported in Loki.

An example rule: `xrule a/1: [age in[18to100],movie_types sim[comedy]]==> [age_filter set union(age_filter,[adult_comedy])]:comedy_rules.` would be interpreted as: for users who are older than 18 and like comedies adjust the *age_filter* attribute and redirect the inference to *comedy_rules* table.

HeaRT (HeKatE RunTime), a dedicated inference engine for the XTT2 rule bases, has been added to Loki as a part of the plugin responsible for parsing Prolog. HMR code is embedded on wiki pages within the `<pl></pl>` tags (see Fig. 2). To run reasoning, a `<pl scope="" goal="">` tag is used. If the goal is a valid HeaRT command, the reasoning is performed by the engine, the result is calculated and rendered on a wiki page. Embedding HeaRT in Loki is currently in an experimental phase and is not provided with the current release.



```
====== Profile page of a User ======
**User age**: 22
<pl>xstat user: [age, 22].</pl>

**User preferences**: horror, comedy
<pl>xstat user: [movie_types, [horror, comedy]].</pl>

<pl>xstat user: [age_filter, []].</pl>
====== Result ======
**The suggested movies are:**

<pl scope="[user|movies]"
    goal="gox(user,[horror_rules,sf_rules,comedy_rules,thriller_rules],tdi),
    print_results.">
</pl>
```

Fig. 2. Goal query on user profile page

4 Summary

In the paper, a semantic wiki called Loki has been presented. An essential feature of the system is a strong rule-based reasoning thanks to a coherent knowledge representation. In the system, standard semantic annotations are mapped to the Prolog knowledge base, in which also rule-based reasoning is specified. Moreover, a custom rule-based engine using decision tables and trees is provided. Loki allows for the development of modularized knowledge bases with the use of a wiki. In future, Loki is planned to be used as a platform for knowledge evaluation [1].

References

1. Baumeister, J., Nalepa, G.J.: Verification of distributed knowledge in semantic knowledge wikis. In: Lane, H.C., Guesgen, H.W. (eds.) FLAIRS-22: Proceedings of the twenty-second international Florida Artificial Intelligence Research Society conference: 19–21 May 2009, Sanibel Island, Florida, USA. pp. 384–389. FLAIRS, AAAI Press, Menlo Park, California (2009)
2. Nalepa, G.J.: PlWiki – a generic semantic wiki architecture. In: Nguyen, N.T., Kowalczyk, R., Chen, S.M. (eds.) Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, First International Conference, ICCCI 2009, Wroclaw, Poland, October 5-7, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5796, pp. 345–356. Springer (2009)
3. Nalepa, G.J.: Architecture of the HeaRT hybrid rule engine. In: Rutkowski, L., [et al.] (eds.) Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010: Zakopane, Poland, June 13–17, 2010, Pt. II. Lecture Notes in Artificial Intelligence, vol. 6114, pp. 598–605. Springer (2010)
4. Nalepa, G.J.: Collective knowledge engineering with semantic wikis. Journal of Universal Computer Science 16(7), 1006–1023 (2010)
5. Nalepa, G.J.: Loki – semantic wiki with logical knowledge representation. In: Nguyen, N.T. (ed.) Transactions on Computational Collective Intelligence III, Lecture Notes in Computer Science, vol. 6560, pp. 96–114. Springer (2011)
6. Nalepa, G.J., Ligęza, A.: HeKatE methodology, hybrid engineering of intelligent systems. International Journal of Applied Mathematics and Computer Science 20(1), 35–53 (2010)