How to Reason by HeaRT in a Semantic Knowledge-Based Wiki

Weronika T. Adrian, Szymon Bobek, Grzegorz J. Nalepa, Krzysztof Kaczor, Krzysztof Kluza
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059
Krakow, Poland
{wta,sbobek,gjn,kk,kluza}@agh.edu.pl

***Abstract*—Semantic wikis constitute an increasingly popular class of systems for collaborative knowledge engineering. We developed Loki, a semantic wiki that uses a logic-based knowledge representation. It is compatible with semantic annotations mechanism as well as Semantic Web languages. We integrated the system with a rule engine called HeaRT that supports inference with production rules. Several modes for modularized rule bases, suitable for the distributed rule bases present in a wiki, are considered. Embedding the rule engine enables strong reasoning and allows to run production rules over semantic knowledge bases. In the paper, we demonstrate the system concepts and functionality using an illustrative example.**

***Keywords*-Knowledge-Based Systems, Knowledge Representation, Reasoning, Semantic Wikis, Knowledge Management**

I. Introduction

The Semantic Web initiative promises new generation of the Web in which machines cooperates with people in solving complex searching, reasoning and planning tasks. Knowledge Representation (KR) methods developed for the Semantic Web stem from research in the field of Artificial Intelligence (AI), but they concentrate more on useful and expressive data representation rather than its intelligent processing. One of the most popular implementations of the Semantic Web concepts are semantic wikis. Over the last few years they have gained a steadily increasing popularity in the fields of knowledge acquisition, management and collaborative knowledge engineering.

Semantic wikis allow for enriching the content with semantic information. The most widely used methods for knowledge representation in semantic wikis are *semantic annotations* popularized by the Semantic MediaWiki (SMW) [**?**] system and its extensions. They allow to assign categories and attributes to wiki pages and define semantic relations among them; they facilitate navigation as well as enable posing semantic queries. The semantically annotated knowledge can be aggregated and queried, and simple classification tasks may be performed. This is useful in encyclopedia-like systems, where users search for data by querying the knowledge included in semantic annotations.

In various practical applications, more flexible and dynamic KR methods are desired. Enriching the representation with rules and providing an inference engine enable reasoning beyond classification and querying. Based on the

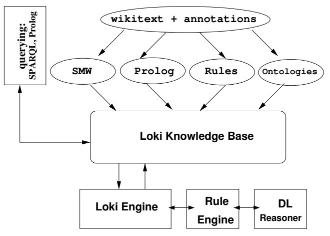


Figure 1. Knowledge representation and processing in Loki-with-HeaRT

information stored in semantic annotations, new facts can be deduced, or desired actions may be performed. Knowledge stored in the system may be considered as a fact base for a rule-based system. In such a system, various inference modes can be considered, including forward and backward chaining inference.

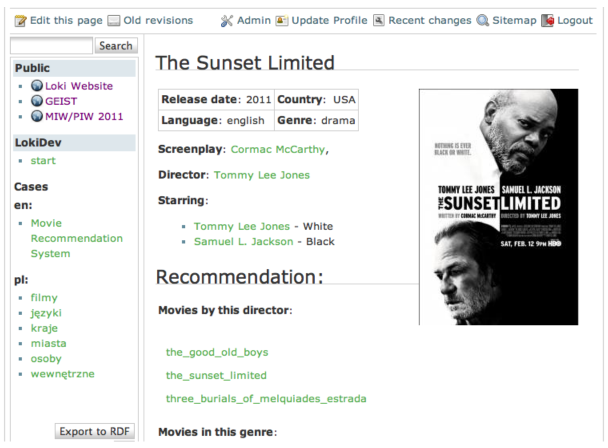
We have integrated a rule engine HeaRT [**?**] with a semantic wiki Loki [**?**] into a hybrid system Loki-with-HeaRT. In this paper, we demonstrate how this combination can be used to develop practical intelligent systems based on semantic annotations and rule-based reasoning. Loki-with-HeaRT combines the semantic wiki flexibility in collaborative knowledge engineering with the power of rule-based representation. Knowledge representation and processing in the system is conceptually shown in Fig. 1. The system most important features can be summarized as follows: 1) It uses a concise logical-based knowledge representation for semantic annotations and rules. 2) It provides reasoning capabilities using powerful Prolog resolution algorithm. 3) By integrating the HeaRT engine, it provides several reasoning modes operating on modularized rule bases. 4) It facilitates knowledge acquisition by keeping compatibility with widely-used semantic annotations. 5) It supports knowledge sharing by exporting the content to RDF and OWL.

We demonstrate the system functionality on a use case of a movie recommendation system. Its main goal is store information about movies and suggest movies to users, based on the information they provide (such as age, favorite genres, authors etc.) as well as the history of seen movies.

The rest of the paper is organized as follows: in Section II we present the knowledge acquisition and querying with semantic annotations in Loki. Rule format and reasoning with the HeaRT engine is outlined in Section III. The underlying knowledge representation and processing of the combined system is then explained in Section IV. The implementation of the system is briefly described in Section V. Summary and future work conclude the paper in Section VI.

II. Basic Semantic Annotations in Loki

The principle objective of Loki is to use a unified logical representation while providing users with diverse knowledge acquisition methods. On one hand, Loki supports semantic annotations as used in SMW, because they are commonly used and intuitive even for untrained users. For example, the information that a movie `A` has been directed by a person `B` can be expressed as: `[[director::B]]` within the wiki page about `A`. Category assignment is done with the `[[category:Name of the category]]` annotation, and an attribute value, e.g. a release date of a movie, can be expressed with `:=` operator (e.g. `[[date:=2011]]`). On the other hand, users can define rules. More experienced users or knowledge engineers can develop a complete rule base within the system (for details see Section III). An example of the wiki page representing a movie and recommendations based on classification is shown in Fig. 2. The source wikitext for this



Public

- Loki Website
- GEIST
- MIW/PIW 2011

LokiDev

- start

Cases

en:

- Movie Recommendation System

pl:

- filmy
- języki
- kraje
- miasta
- osoby
- wewnętrzne

The Sunset Limited

Release date: 2011	**Country**: USA
Language: english	**Genre**: drama

Screenplay: Cormac McCarthy,

Director: Tommy Lee Jones

Starring:

- Tommy Lee Jones - White
- Samuel L. Jackson - Black

Recommendation:

Movies by this director:

the_good_old_boys

the_sunset_limited

three_burials_of_melquiades_estrada

Movies in this genre:

NOTHING IS EVER BLACK OR WHITE.
TOMMY LEE JONES SAMUEL L. JACKSON
THE SUNSET LIMITED
WRITTEN BY CORMAC McCARTHY DIRECTED BY TOMMY LEE JONES
SAT, FEB. 12 9PM HBO

Figure 2. Semantic annotations in Loki

page is as follows:

```
====== The Sunset Limited ======
[[category:movie| ]]
{{ :movie:the-sunset-limited.jpg?w200|}}

| **Release date**: [[date:=2011]] |
  **Country**: [[country:=USA]] |
| **Language**: [[language:=english]] |
  **Genre**: [[genre:=drama]] |

**Screenplay**:
  [[screenplay::person:cormac_mc_carthy
                           |Cormac McCarthy]]
**Director**:
  [[director::person:tommy_lee_jones
                           |Tommy Lee Jones]]
**Starring**:
  * [[starring::person:tommy_lee_jones
                   |Tommy Lee Jones]] - White
  * [[starring::person:samuel_l_jackson
                 |Samuel L. Jackson]] - Black

====== Recommendation: ======
**Movies by this director**:
{{#ask: [[category:movie]]
[[director::person:tommy_lee_jones]]}}

**Movies in this genre**:
{{#ask: [[category:movie]] [[genre:=drama]]}}
```

Adding semantics based on a notion of categories, attributes and relations is straightforward. This facilitates collaborative knowledge engineering, because the wiki markup is flexible and easy to learn. In the presented use case, users can add new movies, describe them with attributes and connect with relations, extend the movie and people descriptions collaboratively developing the system knowledge base.

Semantic annotations allow for querying the wiki with use of a simple `ask` syntax as in SMW. Within a query one can use logical operators, wildcards, subqueries and property chains. Moreover, the sub-class and sub-properties relations are analyzed and the simple reasoning based on classification is performed. The output of the queries may be formatted as a list, a table, a CSV file etc. (see the documentation[1]).

Loki supports Semantic Web standards for querying and exporting the system data. SPARQL syntax is supported with an assumption that only the wiki system may be queried, not any external source. The semantic content of the wiki may be exported to an RDF/XML file (see the Export button in Fig. 2). Wiki pages are exported as instances, relations as OWL Object Properties, attributes as Datatype Properties. Subcategories are interpreted as OWL subclasses.

III. Reasoning with HeaRT

HeKatE RunTime (HeaRT) [**?**] is a lightweight embeddable rule inference engine developed in the HeKatE project [**?**] [2]. It uses an expressive visual rule representation language called Extended Tabular Trees (XTT2) [**?**] which is based on a network of connected decision tables. An exemplary network is given in Fig. 3. The XTT2 representation allows HeaRT to support two inference modes: Data Driven inference (DDI) and Goal Driven inference (GDI) which are forward and backward chaining algorithms.

Visual representation of the rule base is human-readable but difficult to parse and process. To this end, the HMR rule language is used [3] It is a simple and readable text representation suitable for automated processing. The textual representation is automatically generated from the visual one, by a dedicated design tools [**?**].

[1]See http://loki.ia.agh.edu.pl.
[2]See http://hekate.ia.agh.edu.pl.
[3]See http://ai.ia.agh.edu.pl/wiki/hekate:hmr.

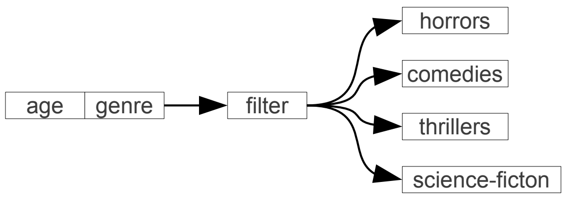


Figure 3. Schema of the XTT2 visual representation of the system

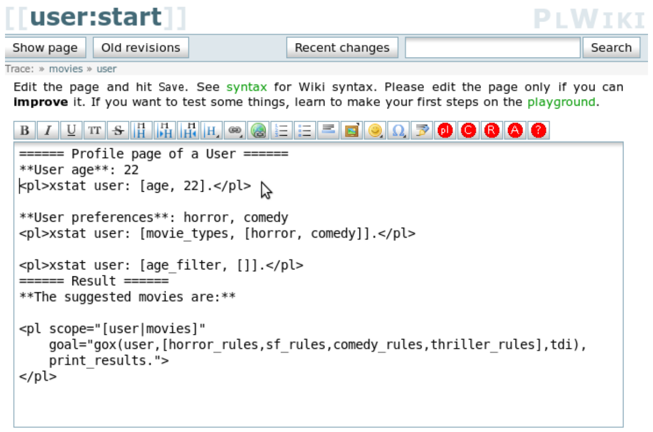
Let us now demonstrate rules used in the recommendation system outlined in previous section. Rules written in HMR would be added and used to recommend a movie set for a user of a given age and some film genre preferences. We use separate wiki namespaces for each user of the system and another namespace for movies. The schema of a visual representation of the rule base is presented in Fig. 3.

Let us analyze the DDI run over this knowledge base. In the first step, the system decides for which subsets of movies a user is allowed based on his age. For instance, users younger than 18 are not allowed to watch horrors and thrillers nor other movies that age limit is higher than 18. Then, based on the age filter, the system search for movies that best fit user preferences specified in his profile. In the end, the system responds with a list of recommended movies.

HMR language is embedded on wiki pages within `<pl></pl>` tags. An exemplary rule written in HMR is shown below. The rule should be read as follows: *If the age of the user is less then 18 and he marked horrors and thrillers as his preferred genres, then set age filter to none.*

```
xrule filter/1:
  [age lt 18,
   movie_types sim [horror, thriller]]
    ==> [age_filter set [none]].
```

To initialize attribute values (user age and preferences), the *xstat* element from HMR language is used (see Fig. 4).



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.

```
====== 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>
```

Figure 4. Movie recommendations on a user profile page

In the presented case rules are located on separate pages. Such a modularization of knowledge base is especially useful when there may be several different sets of rules that have common parts. It is easy to control system goals by controlling modules (namespaces) with rules. To merge all information, a scope has to be given when a goal for the inference is specified. In the discussed system, the construction `scope="[user|movies]"` is used.

It is also possible to browse through the database of movies and get information whether selected movie is recommended for the user or not. This is a different reasoning task, because the inference works backward, not forward. What is important, this functionality can be achieved without modifying the rule base. Technically speaking, only one line of the wiki markup must be changed, e.g.:

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

IV. Knowledge Representation and Processing

The knowledge acquisition layer described in previous sections covers the semantic annotations as used in Semantic MediaWiki (see Sect. II) as well as rules and state definitions in HMR language (see Sect. III). All the knowledge, represented in various ways, upon saving the wiki page is translated to the underlying logical representation. This representation is based on a Prolog subset of First Order Logic which allows to formulate and process Horn clauses.

Semantic annotations are translated into facts as follows:

```
[[category:movie]]
 wiki_category('movie','the_sunset_limited').

[[director::person:tommy_lee_jones
 wiki_relation('the_sunset_limited',
   'director','tommy_lee_jones').

[[date:=2011]]
 wiki_attribute('the_sunset_limited',
   'date','2011').
```

Semantic queries are mapped into Prolog goals as:

```
{{#ask: [[category:movie]]
   [[director::person:tommy_lee_jones]]}}
 wiki_category('movie',Page),
 wiki_relation(Page,'director',
   'tommy_lee_jones').
```

To answer the semantic queries, the core Loki engine realizes the goals against the knowledge base, using the Prolog resolution algorithm. For more advance inference, the HeaRT engine is used. It can directly operate on facts stored in the KB resulted from the semantic annotations.

One can also embed Prolog code (facts and rules) into the wiki, which may be a convenient way of knowledge engineering for a group of qualified users. Nevertheless, it is not necessary to know the Prolog syntax – using simple annotations and HMR rule language serves the goal of Loki to provide a flexible and user-friendly environment.

V. System Design and Implementation

Loki has been designed as an extension to a popular wiki engine Dokuwiki [4]. The Loki functionality has been added with used of Dokuwiki plugins mechanism. The first prototype implementation called PlWiki (Prolog-based Wiki) has been described in [**?**]. Knowledge engineering with this prototype has been discussed in [**?**]. For details about the system architecture see [**?**].

Technical aspects of the integration of the core Loki engine with HeaRT has been described in [**?**]. The architecture of Loki-with-HeaRT is divided into two modules: the one responsible for rendering wiki pages and extracting the HMR code, and the other – for performing inference based on the HMR model passed to it by the Loki engine.

The process of rendering a wiki page in Loki with HeaRT is as follows: 1) Wiki engine parses the wiki page and extracts semantic information (categories, relations, attributes, HMR code) and reasoning queries or goals for HeaRT. 2) If HMR code for HeaRT is present, then depending on a scope defined in the goal, Loki merges the HMR code from wiki pages in a given scope and passes it to HeaRT. 3) HeaRT performs the reasoning process and returns results to the Loki engine. 4) Loki renders complete wiki page the answer to a given query or goal.

VI. Summary

Loki is a knowledge-based semantic wiki that combines flexible knowledge representation with strong reasoning mechanisms. On one hand, it is compatible with popular semantic annotations. On the other, it is integrated with a rule engine able to operate in various modes over modularized rule bases. The unified underlying representation enables processing knowledge acquired in different ways. The architecture of the system is modularized and expandable.

The current state of semantic wikis development can be traced on the `semanticweb.org` portal [5]. Most of semantic wiki systems that introduce rules to their syntax, do not take full advantage of this representation. For instance, rules available in AceWiki [**?**] are translated to OWL, and during reasoning are processed as First-Order calculus expressions loosing strength of rule representation. To the best of our knowledge, no structuralization of knowledge base is provided by any of the semantic wiki systems. In KnowWE [**?**], there is a possibility of expressing knowledge with decision tress, but during translation from decision trees to rules, the dependencies between them are lost, which slows down the inference process.

For future work, we plan to implement an import facility, so that external ontologies can be used in the system. Moreover, integration with HeaRT will move from embedding the engine into communication over TCP/IP network. Using this architecture, more information will be exchanged between Loki and HeaRT, e.g. the logged in user's identifier, the currently viewed wiki page, the date and possibly more state-related information. This will allow for even more flexible and dynamic operations based on knowledge in the system.

Acknowledgment

The paper is supported by the AGH UST Grants.

References

[4]See http://dokuwiki.org.

[5]See http://semanticweb.org/wiki/Semantic_wiki_projects.