

Dynamic Queries in Semantic Wiki Systems

Mirko Čubrilo, Markus Schatten, Jurica Ševa

Faculty of Organization and Informatics

University of Zagreb

Pavlinska 2, 42000 Varaždin, Croatia

{mirko.cubrilo, markus.schatten, jurica.seva}@foi.hr

Abstract. *Semantic wiki systems have gained major attention of the academic community in the past few years. The idea of integrating the semantic web with fast and lightweight content management systems seems to be good direction towards web 3.0 or the social semantic web. Herein we analyze how to take advantage of such systems through dynamic querying facilities. Particularly we analyze the τ AOPIs system which has a build in frame logic reasoning engine, and show how queries can enhance the presented content through dynamically generated class hierarchies, dictionaries, FAQ-s, tables of content, issue tracking and others. We conclude that by enriching wiki syntax with dynamic queries semantic wikis can gather additional functionality in organizing knowledge.*

Keywords. semantic wiki; dynamic queries; frame logic; taopis

1 Introduction

Wiki systems, a progressive technology that hasn't been predicted a bright future by prominent scientists, are in wide usage today. These systems, that allow any user to add different content to the system, are autopoietically evolving into more and more impressive knowledge repositories. Maybe the best known example of such systems, Wikipedia, the free Internet encyclopedia, had over 2.2 million articles in its English version at the time of writing this text, whilst there are versions in almost all world languages.

Still, it seems that wiki systems came to their edge [11]. It is more and more often the case that various rules concerning behavior, knowledge orga-

nization as well as metadata are implemented in order to facilitate search and reasoning in these often huge (predominant textual) data repositories [2, 6, 7, 4].

As argued in [9] semantic wiki systems are an extension to common wiki systems that use additional functionality borrowed from the semantic web. In order to ease search, integration and reasoning, users can provide metadata, which yields additional semantics or meaning [10].

The τ AOPIs system, as presented in [1, 8, 10, 5, 9] has a semantic wiki subsystem that is based on frame logic [3], and particularly uses the \mathcal{F} LORA-2 reasoning engine [12] to allow its users to query the dynamically created knowledge base. τ AOPIs uses a syntax entitled **niKlas** that comprises the possibility to cast dynamic queries inside any wiki page.

Herein we show how such dynamic queries can be used to generate new and useful content, using the previously mentioned **niKlas** syntax. The queries are dynamic in terms that their results are always up to date with the dynamically created knowledge base.

2 τ AOPIs Object – Oriented Framework

τ AOPIs is an open source social web application that aims to provide a platform for self-organizing communities. Such communities can be either organizations or projects for which τ AOPIs provides suitable tools like semantic wiki systems, forums, blogs, ranking mechanisms, content filtering, tagging etc.

Its unique feature is to approach any content (forum posts, blog entries, blog comments, and particularly wiki pages) as objects in the sense of the object-oriented paradigm. Such objects can have relations to other objects (manifested in hyperlinks), attributes (manifested through attribute-value tags, provided through a social tagging system), methods (manifested in web services), and can be classified into user-defined classes (manifested through a special case of attribute-value tags).¹

This object-oriented approach to semantic wiki systems, allows us to query any wiki page (or object in a boarder sense) for its class, attribute names, attribute values, relation names, related objects as well as methods. Additionally **τΑΟΠ̄IS** allows the querying of authorship (users also represent objects), project or organization membership and provides sorting facilities.

3 niKlas Syntax for Dynamic Queries

The **niKlas** syntax, which is similar to phpBB code², defines beside different text formatting commands, a query command:

```
[query flora2_query . ]
{{header}header_formatting[/header]}
answer_formatting
[/query]
```

whereby *flora2_query* is a normal (restricted) **FLORA-2** query with defined return variables, *header_formatting* is the optional header (possibly formatted using **niKlas** code),³ and *answer_formatting* is a **τΑΟΠ̄IS** formatting that can contain variables used in the **FLORA-2** query. The *answer_formatting* is repeated for any answer returned by the **FLORA-2** reasoning engine by using the generated **FLORA-2** ontology of the semantic wiki as a knowledge base.

¹For a better understanding of **τΑΟΠ̄IS** semantics please refer to [9]

²phpBB is a free open-source forum and messaging board system available at <http://www.phpbb.net>

³In the current version subqueries are not allowed.

4 Use Cases

In the following a few interesting use cases of dynamic queries shall be analyzed. We will show how **niKlas** can be used to dynamically generate new content that is usable in a wide range of community projects like current class hierarchies, dictionaries, FAQ's, tables of content, lists of editors, issue and bug tracking, as well as what links here links.

4.1 Class Hierarchies

In order to obtain a dynamically generated list of classes with corresponding subclasses one can issue the following query:

```
[query ?sub::?super . ]
?sub is a subclass of ?super
[/query]
```

The first line defines the **FLORA-2** query which is in particular a class expression with variables. The second line define the formatting for the given list (which of course could have been a table or any other formatting). The third line closes the formatting.

Such a query will yield a list similar to:

```
apple is a subclass of fruit .
banana is a subclass of fruit .
fruit is a subclass of food .
```

Such a query would be usefull if using a semantic wiki system as an ontology management tool, to provide a detailed overview of existing classes with corresponding subclasses. There are ofcourse variations to this query that would allow to obtain only strict subclasses, subclasses of a special class etc.

4.2 Dictionaries

Another interesting feature that can be provided using a dynamic query is a dictionary. Some wiki systems provide such a facility as a build-in function. Herein we show how such a functionality can be simulated in **niKlas** through a serious of queries similar to the following, provided that any wiki page is tagged with its corresponding first letter.⁴

⁴This tag is needed due to the fact that **FLORA-2** syntax is a bit complex when it comes to string processing. A better

```
[ query ?_ : wiki_page [
  title ->?title ,
  url ->?address ,
  letter ->A ] ,
  sort (?title , asc) . ]
[ header ] [ h1 ] A [ / h1 ] [ / header ]
[ link =?address >?title ]
[ / query ]
```

The query if provided on some wiki page would yield a result similar to the following whereby the corresponding titles would be links to the particular wiki pages dealing with them.

```
[ h1 ] A [ / h1 ]
[ link =AnanasPage >Ananas ]
[ link =ApplePage >Apple ]
...
```

To provide a full dictionary at the current version of **niKlas** the user has to make a query for any letter of a given alphabet. This is of course a unpleasant solution that implies that some future version of **niKlas** has to provide a facility for subqueries and functionality similar to the SQL *group by* clause.

Such a query is usable in almost any wiki system. Especially in encyclopedia-like wikis such a query can be of great value.

4.3 Frequently Asked Questions

The popular FAQ section of some project or service can be simulated using a dynamic query similar to the following provided that any wiki page that is an actual answer to a frequently asked question is tagged with attribute *question* and the corresponding question as its value.

```
[ query ?_ : wiki_page [
  question ->?question ,
  url ->?address ] ,
  sort (?question , asc) . ]
[ link =?address >?question ]
[ / query ]
```

The query would yield a sorted list of frequently asked questions with links to their answers similar to:

solution using wildcards or regular expressions should be provided in some future version of **niKlas** .

```
[ link =ApplePie >What is apple pie? ]
[ link =PineApple >What is pineapple? ]
...
```

Queries similar to this can be useful in almost any project or organization that interacts closely with their users/customers. Especially open source projects, customer relationship sites as well as others could have considerable benefits.

4.4 Tables of Content

In order to provide a book-like “linearization” of a wiki site one could provide a table of content using a series of queries similar to the following, provided that wiki pages are tagged with the chapter they belong to.

```
[ query ?_ : wiki_page [
  chapter ->KM,
  title ->?title ,
  url ->?address ] ,
  sort (?title , asc) . ]
[ header ] Knowledge management [ / header ]
[ link =?address >?title ]
[ / query ]
```

Such a query would provide a list of sections of a given chapter with hyperlinks to the corresponding pages.

```
Knowledge management
[ link =Explicit >Explicit knowledge ]
[ link =Tacit >Tacit knowledge ]
...
```

In this case again **niKlas** current version shows a drawback for not supporting subqueries, since the user has to issue a query for any chapter.

Such a query (or series of queries) is interesting in any documentation project, thesis, on-line book etc.

4.5 Who Edited this Page

One sometimes would like to know which users contributed to a given page. In **niKlas** this is achieved to a bit more complex query as follows.

```
[ query ?_ : wiki_page [
  title ->'Page title ' ,
```

```

    author->?_a ],
    ?_a: person [
        name->?name,
        surname->?surname,
        email->?email ]. ]
[ header ] Contributors:[ / header ]
?name ?surname
[ url ] mailto:?email[ / url ]

[ / query ]

```

By issuing this particular query we would obtain a list of users' names and surnames with links to their e-mail addresses.

```

Contributors :

Mirko Cubrilo
[ url ] mailto:mcubrilo@foi.hr [ / url ]

Markus Schatten
[ url ] mailto:mschatte@foi.hr [ / url ]

Jurica Seva
[ url ] mailto:jseva@foi.hr [ / url ]

```

Such queries are useful in any wiki site that wants to keep track of contributors to (for instance) facilitate the creation of a social network.

4.6 Issue Tracking

Lots of projects (especially information system based projects) have a particular need to keep track of issues that arise during the project (bugs, feature requests etc.). Provided that any wiki page that holds a bug description is tagged with *class:bug* as well as with *status:open* the following query would yield a list of open bugs.

```

[ query ?_ : bug [
    status->open ,
    not( status->closed ) ,
    title->?title ,
    url->?address ] ,
    sort(?title , asc). ]
[ header ] Open bugs [ / header ]
[ link=?address>?title ]
[ / query ]

```

The list would look similar to the following:

```

Open bugs
[ link=BuildBug>Build crashes ]
[ link=ErrorOpen>Error on open ]
...

```

In order to close an issue, a project member that solved it, just needs to tag the wiki page with *status:closed*, and the issue would be removed from the list. As indicated above, such a dynamic query can be useful on any project related site that deals with any kind of issues that one needs to keep track of.

4.7 What Links Here

A feature often included into conventional wiki systems is a list of pages that link to the current page. Provided that '*Page title*' is the title of the current page, this functionality can be simulated in niKlas as follows.

```

[ query ?_ : wiki_page [
    title->?title ,
    url->?address ,
    ?_->?_this_page ] ,
    ?_this_page: wiki_page [
        title->'Page title' ]. ]
[ header ] What links here? [ / header ]
[ link=?address>?title ]
[ / query ]

```

Thereby the result to this query would be a list of page titles with corresponding hyperlinks to the current page including links to them.

```

What links here?
[ link=BananaPage>Banana ]
[ link=OrangePage>Orange ]
...

```

This functionality is useful especially on wiki sites that have complex mutual hyperlinks (encyclopedia-like wikis, technical documentation etc.).

5 Conclusion

Semantic wiki systems with dynamic querying facilities are useful in almost any knowledge management environment where summary information

is needed. The paper showed how dynamic queries can be used to generate additional and usefull content from the knowledge base of a semantic wiki system. Particularly by analyzing the $\tau\text{AOP}\bar{\text{I}}\text{S}$ system and its **niKlas** syntax we were able to show that one can gather various summary information from the knowledge base including class hierarchies, dictionaries, frequently asked questions, tables of content, contributor lists, issue tracking lists as well as lists of pages that link to the current one. Such facilities are of value to different project and organization related situations.

niKlas showed a few drawbacks in its current version due to the impossibility to issue subqueries, use grouping in queries as well as wildcards nor regular expressions. These and other issues should be addressed in future versions of the language. On the other hand, due to the strong formalism of frame logic in its background, **niKlas** proved to solve interesting object-oriented tasks as shown herein.

References

- [1] M. Bača, M. Schatten, and D. Deranja. Autopoietic information systems in modern organizations. *Organizacija, Journal of Management, Informatics and Human Resources*, 40(3):157–165, 2007.
- [2] S. E. Campanini, P. Castagna, and R. Tazzoli. Platypus wiki: a semantic wiki wiki web. In *Semantic Web Applications and Perspectives (SWAP) 1st Italian Semantic Web Workshop*, Ancona, Italy, 10th December 2004 2004.
- [3] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the Association for Computing Machinery*, 42:741–843, May 1995.
- [4] M. Krötzsch, S. Schaffert, and D. Vrandečić. Reasoning in semantic wikis. In *Reasoning Web Summer School*, Dresden, September 2007. Springer-Verlag.
- [5] M. Maleković and M. Schatten. Leadership in team based knowledge management - an autopoietic information system's perspective. In B. Aurer and M. Bača, editors, *19th Central European Conference on Information and Intelligent Systems – CECIIS2008 Conference Proceedings*, pages 47–52. Faculty of Organization and Informatics, September 2008.
- [6] E. Oren, M. Völkel, J. G. Breslin, and S. Decker. Semantic wikis for personal knowledge management. In *International Conference on Database and Expert Systems Applications (DEXA) Proceedings*, pages 509–518, 2006.
- [7] S. Schaffert. Ikwiki: A semantic wiki for collaborative knowledge management. In *1st International Workshop on Semantic Technologies in Collaborative Applications STICA 06 Proceedings*, Manchester, UK, 2006.
- [8] M. Schatten, J. Brumec, and M. Višić. Strategic planning of an autopoietic information system. In B. Aurer and M. Bača, editors, *Proceedings of the IIS 2007 18th International Conference on Information and Intelligent Systems*, pages 435–440. Faculty of Organization and Informatics, 2007.
- [9] M. Schatten, M. Čubrilo, and J. Ševa. A semantic wiki system based on f-logic. In B. Aurer and M. Bača, editors, *19th Central European Conference on Information and Intelligent Systems – CECIIS2008 Conference Proceedings*, pages 57–61. Faculty of Organization and Informatics, 2008.
- [10] M. Schatten and M. Žugaj. Organizing a fishnet structure. In *29th International Conference Information Technology Interfaces Proceedings*, pages 81–86, Cavtat-Dubrovnik, Croatia, June 25 — 28 2007.
- [11] M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic wikipedia. In *International World Wide Web Conference (IW3C2), WWW 2006 Proceedings*, Edinburgh, Scotland, May 23—26 2006.
- [12] G. Yang, M. Kifer, and C. Zhao. Flora-2: A rule-based knowledge representation and inference infrastructure for the semantic web. In *Second International Conference on Ontologies, Databases and Applications of Semantics (ODBASE) Proceedings*, Catania, Sicily, Italy, November 2003.