Semantic Mashup with the Online IDE WikiNEXT

Pavel Arapov, Michel Buffa, Amel Ben Othmane

To cite this version:

Pavel Arapov, Michel Buffa, Amel Ben Othmane. Semantic Mashup with the Online IDE WikiNEXT. Demo session. 2014, pp.4. <10.1145/2567948.2577010>. <hal-01083331>

HAL Id: hal-01083331

https://hal.inria.fr/hal-01083331

Submitted on 18 Nov 2014

Semantic Mashup with the Online IDE WikiNEXT

Pavel Arapov, Michel Buffa, Amel Ben Othmane
Wimmics research group
INRIA and I3S Laboratories
Sophia Antipolis, France
arapov@i3s.unice.fr, buffa@i3s.unice.fr, amel.ben-othmane@inria.fr

ABSTRACT

The proposed demonstration requests DBPedia.org, gets the results and uses them to populate wiki pages with semantic annotations using RDFaLite. These annotations are persisted in a RDF store and we will show how this data can be reused by other applications, e.g. for a semantic mashup that displays all collected metadata about cities on a single map page. It has been developed using WikiNEXT, a mix between a semantic wiki and a web-based IDE. The tool is online[1], open source[2]; screencasts are available on YouTube (look for "WikiNext").

Categories and Subject Descriptors

K.4.3 [**Organizational Impacts**]: Computer-supported collaborative work.

Keywords

Semantic Web, Web2.0, Wikis, Semantic Wikis, Knowledge Management, Web Applications

1. Introduction / motivation

Wikis appeared in 1995 with Ward Cunningham's wiki, changing radically the way people could produce documents on the web. It was the beginning of the web 2.0. Soon many different wiki engines (this is how we call the software that powers wiki web sites) appeared, specialized for writing online encyclopedias (MediaWiki, that powers Wikipedia), or targeted to private organizations [5]. We call these ones "application wikis", they propose features, such as: database integration, access control, versioning, possibility to embed HTML forms in pages, templates for formatting documents, and the possibility to write small applications using the page editor. Wiki engines such as TWiki, FOSWiki, TikiWiki, Confluence, XWiki, Mindtouch, belong to that category of wikis. In 2005, researchers from the Semantic Web domain introduced the "semantic wikis": wikis for creating/maintaining ontologies (data models) collaboratively and/or for annotating wiki content with semantic metadata, and enable requesting/reasoning on the embedded data. Overviews of such wikis can be found in [5]. Our group has a strong experience in that field as we developed two semantic wikis over the last seven years. We also wrote for the W3C[3] an HTML5 course[4] that includes lots examples that involve HTML/JavaScript/CSS. This course is powered by the moodle[5] e-learning platform; a CMS specialized for managing online courses of any kind. Integrating interactive programming examples in moodle turned to be so painful and limited (using, iframes) that we hosted externally the examples on jsbin.com, a popular web-based IDE. Our research group also developed the French version of DBPedia.org[6], a semantic database that holds the data from Wikipedia and accepts SPARQL[7]. This project needed a front end for developing examples programs that consume and exploit its data. Here, too, we imagined a wiki engine that could act as a front-end: one could write applications from within the wiki, which can talk to DBPedia.fr, could write tutorials made both of text and application code that can be modified collaboratively. These ideas led to WikiNext, a wiki engine that integrates HTML5/JavaScript, proposes some APIs for exploiting the web of data (request and process data from different sources, cache them, aggregate them, write code that manipulate them for visualization, mashups, annotating text content, etc.). WikiNEXT mixes the editing of "documents" and the editing of "source code of applications within the documents", provides a templating system, comes with examples, tutorials, has features for adding external JavaScript libraries, etc.

2. State of the art

Semantic Wikis: The first wave of semantic wikis started in 2005, with wiki engines such as Semantic Media Wiki [1], IkeWiki [2] OntoWiki [4] or SweetWiki [3]. They all proposed to add semantic metadata to documents, and used internally some formal language to store them, such as RDF (Resource Description Framework), OWL (Web Ontology Language), or Conceptual Graphs [7]. These metadata can be exploited to add new functionalities to the wiki: augmented navigation, visualization of metadata, search and retrieval or reasoning. The SPARQL language[8] is often used internally for querying RDF data. Wiki engines like Semantic MediaWiki based on the MediaWiki engine, or SemperWiki[8] used *wiki markup syntax* for adding semantic annotations to wiki pages. OntoWiki is more of an ontology editor, and proposes user-friendly interface that offers different views and browsing and editing interfaces over existing data.

Today, semantic wikis, which are still active [15] either (1) kept adding new functionalities using plugins or extensions or (2) relied on external tools or web services. The Halo[9] extension of Semantic Media Wiki proposes forms, auto-completion, a WYSIWYG editor, the integration of audios and videos files, and the integration of a SPARQL endpoint, or more complex

[1] http://wikinext.gexsoft.com

[2] https://github.com/pavel-arapov/wikinext

[3] See http://www.w3devcampus.com

[4] http://w3devcampus.com

[5] https://moodle.org/

[6] http://DBPedia.fr

[7] http://www.w3.org/TR/rdf-sparql-query/, the W3C standard for querying the web of data.

[8] http://www.w3.org/TR/sparql11-query/

[9] http://www.projecthalo.com/

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577010

extensions such as MoKi [13] (focuses on enterprise modeling) and OWiki [12] (ontology-driven generation of wiki content), also based on Semantic Media Wiki, belong to the first category. The KiWi project (Knowledge in Wiki [10]), successor of IkeWiki, or SweetDeki [11] the wiki that succeeded to SweetWiki[3], and has been integrated in the ISICIL ANR project [11], belong to the second category.

Web-Based IDEs: JavaScript based source code editors like Code Mirror[10], ACE Cloud9[11] or ternjs[12] propose features such as syntax highlighting, auto-completion, and have been used by many online IDEs, well known by *web developers*, such as jsbin.com, jsfiddle.net or tinkerbin.com. These IDEs enable real time editing of HTML/CSS/JavaScript code in the browser, instant preview when files are updated, etc. Most of these tools however, are sandboxes for testing code online or for writing small examples like the ones we developed for the HTML5 course we wrote. WikiNext shares lots of the features found in the last generation of these tools (instant preview, run code in the browser, etc.) and mixes them with a wiki approach, an API for exploiting Linked Data and an original backend that includes an RDF data store and a NOSQL database. Figure 1 shows the code editors present in the WikiNEXT user interface.

3. WikiNEXT

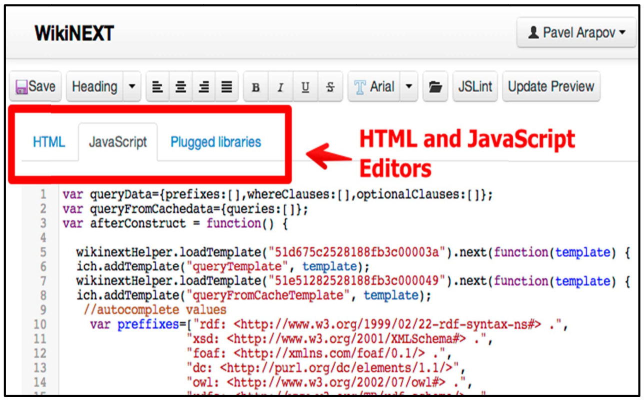
Application wikis, semantic wikis, modern online IDEs: we tried to mix them all with WikiNEXT; *we considered each page as a web application*, composed of HTML/CSS/JavaScript code. We provided several editors in the browser, for editing the text and the JavaScript source code, and provided an API for "talking with the wiki engine internals". Consider the text content of a wiki page as data: the application part of the page (JavaScript code) can collect semantic data from an external source, use them for annotating the document, extract information from the text for suggesting tags, add graphic visualizations, save data in a cache, etc. WikiNEXT has been designed for making the process of writing such application easier, and without the need to use any extra tool. It proposes a real client side application layer that enables richer data manipulation using semantic web standards, an API for caching and updating data, and more.

The online prototype comes with many examples and tutorials, e.g. for writing applications that exploit the semantic database DBPedia.org, for processing the semantic data collected and enhancing wiki documents, etc. This makes WikiNEXT a good tool for teaching semantic web technologies, without the need to install any software. It is also a good tool for collecting data and integrating them in documents.

3.1 Architecture overview

The WikiNEXT JavaScript API provides numerous objects, events and methods that can be used by applications developed within the wiki. In an MVC paradigm, the JavaScript part of the page will be the business layer and the Controller, the View part will be done in the HTML/CSS of the page, and the Model part is composed of RDF metadata.

WikiNEXT articles/pages are associated with a set of metadata that describe their basic characteristics: Title, Article, Author, Contributor, Date, etc. But articles are also containers: they can hold metadata created manually or added automatically by an application. WikiNEXT uses schema.org[13] vocabularies to describe the default structure of articles, users' details, etc. Each article is represented semantically as a named graph made of RDFa/RDFalite annotations, whose name is based on its URI (Uniform Resource Identifiers). We call this graph the *Local Knowledge Base*. The full stack of wiki articles give us the *Global Knowledge Base*, the union of all the named graphs.



```
var queryData={prefixes:[],whereClauses:[],optionalClauses:[]};
var queryFromCachedata={queries:[]};
var afterConstruct = function() {

  wikinextHelper.loadTemplate("51d675c2528188fb3c00003a").next(function(template) {
  ich.addTemplate("queryTemplate", template);
  wikinextHelper.loadTemplate("51e51282528188fb3c000049").next(function(template) {
  ich.addTemplate("queryFromCacheTemplate", template);
   //autocomplete values
    var preffixes=["rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .",
                   "xsd: <http://www.w3.org/2001/XMLSchema#> .",
                   "foaf: <http://xmlns.com/foaf/0.1/> .",
                   "dc: <http://purl.org/dc/elements/1.1/>",
                   "owl: <http://www.w3.org/2002/07/owl#> .",
```

Figure 1: WikiNEXT page editors, the different tabs show the HTML editor, the JavaScript editor (current view) and the library management tab.

The metadata generated by applications embedded in the page/article, e.g. a list of countries resulting from a request or a web services call to DBPedia.org, are also represented in the page as RDFa/RDFaLite[14] annotations. WikiNEXT parses these annotations and saves them in a modified version of RDFStore-js [14], which supports SPARQL 1.1. The traditional page content is stored as objects in a MongoDB[15] database, and indexes for the textual page content are generated this way.

3.2 Implementation

WikiNEXT has been written from scratch in JavaScript and relies on the Node.js HTTP server and on the MongoDB database. Node.js[16] (Node) is an I/O environment built on top of Google Chrome's JavaScript runtime — essentially, a server-side implementation of a JavaScript interpreter. MongoDB is a No SQL document-oriented database. RDFstore-js triples are also persisted in MongoDB, then, RDF triples are both accessible through RDFstore-js using SPARQL, but also directly through MongoDB. With these technological choices, we tried to minimize the number of languages and the number of data formats used in WikiNEXT: JavaScript can be used in the whole process of development both for programming and for persistence, both on the client side and on the server side.

4. Demonstration proposed: the City Application

This application requests DBPedia.org, gets the results and uses them to populate wiki pages with semantic annotations using RDFaLite. These annotations are persisted in the wiki RDF store and we will show how this data can be reused by other

[10] http://codemirror.net/

[11] http://ace.c9.io/

[12] http://ternjs.net/

[13] http://schema.org/ and http://schema-rdfs.org have been created by the main search engine vendors and propose several RDF/S schemas for representing persons, articles, etc.

[14] http://www.w3.org/TR/xhtml-rdfa-primer/ and http://www.w3.org/TR/rdfa-lite/

[15] http://mongodb.org

[16] http://nodejs.org

applications, e.g. for a semantic mashup that displays all collected metadata about cities on a single map page. Screencasts of this application are available on YouTube (look for "WikiNext").

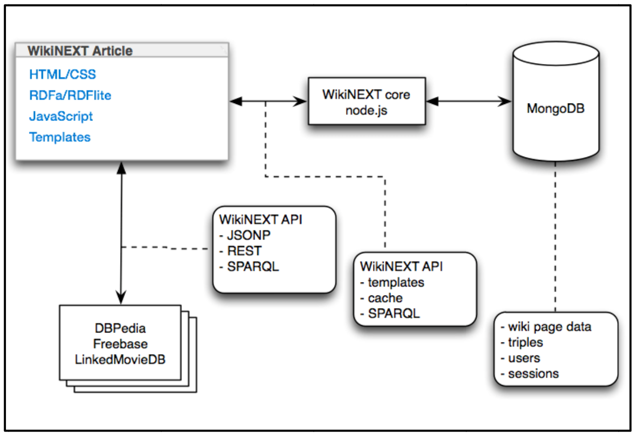


Figure 2: WikiNEXT Architecture.

This application is online on the WikiNEXT web site[17] and can be run, edited or cloned by any user. It is composed of:

1. A View form: as shown in Figure 3, it allows users to look for information about a city and creates a new page with the found information or makes an update if the page for this city already exists in the wiki.
2. An application code that finds information about a city by querying DBPedia.org[18]'s SPARQL endpoint.
3. A template for displaying the data collected, we use a template that gives a nice layout.
4. Some code for creating a new page: in order to reuse information, WikiNEXT offers the possibility to create pages that contain both information about cities and semantic annotations.

4.1 Retrieving data

The WikiNEXT API proposes some "default" rendering for displaying RDF data, we call this mode "sparql On Fly", see the example of Figure **4**. On the other hand, it is possible to create some "template pages" in order to have a more meaningful and structured view of the information, as shown in Figure 5.

The principle is simple to use and permits the decoupling of the UI (User Interface) definition from the data. The user needs first to define the UI in the HTML part as the example below:

```
Welcome to {{city}}.
```

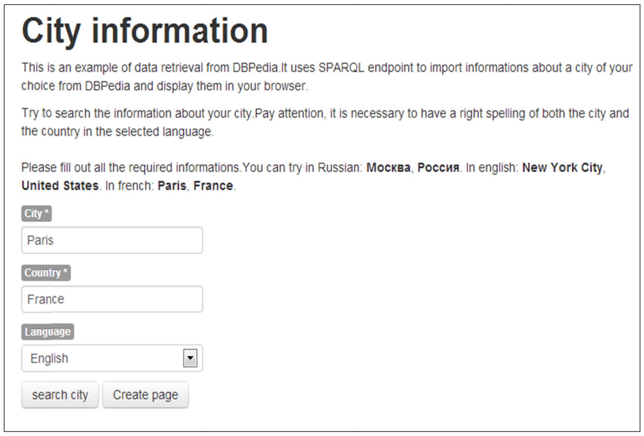
In the JavaScript part data will take this form:

```
var data = {city: 'Nice'};
```

The result will be*:*

```
Welcome to Nice.
```

WikiNEXT proposes an API with various features to cache the data collected from external data sources, refresh existing ones, etc. Figure 2 shows the complete functional diagram of the WikiNEXT architecture.



City information

This is an example of data retrieval from DBPedia.It uses SPARQL endpoint to import informations about a city of your choice from DBPedia and display them in your browser.

Try to search the information about your city.Pay attention, it is necessary to have a right spelling of both the city and the country in the selected language.

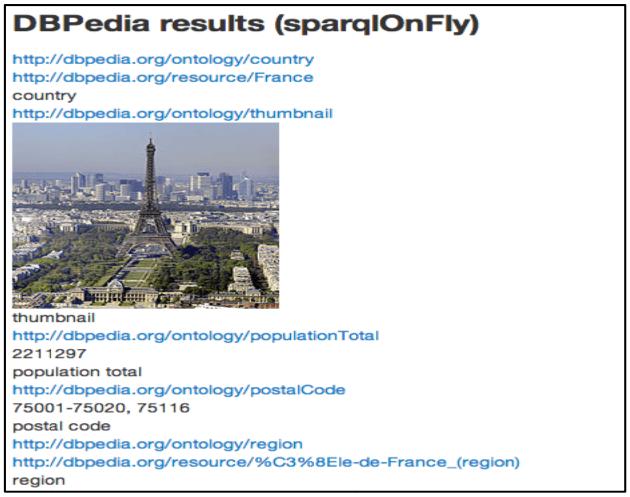
Please fill out all the required informations.You can try in Russian: **Москва**, **Россия**. In english: **New York City**, **United States**. In french: **Paris**, **France**.

City *
Paris

Country *
France

Language
English

Figure 3: City application, here we see the form that will request DBPedia.org and display the results in the local page or in a new page.



DBPedia results (sparqlOnFly)

http://dbpedia.org/ontology/country
http://dbpedia.org/resource/France
country
http://dbpedia.org/ontology/thumbnail
thumbnail
http://dbpedia.org/ontology/populationTotal
2211297
population total
http://dbpedia.org/ontology/postalCode
75001-75020, 75116
postal code
http://dbpedia.org/ontology/region
http://dbpedia.org/resource/%C3%8Ele-de-France_(region)
region

Figure 4: Results from DBPedia, default rendering without using a template. We call this mode "sparql On Fly".

4.2 Processing data

The template both handles the display of the data retrieved from DBPedia, and the RDFa Lite annotations. For example, a city is defined by:

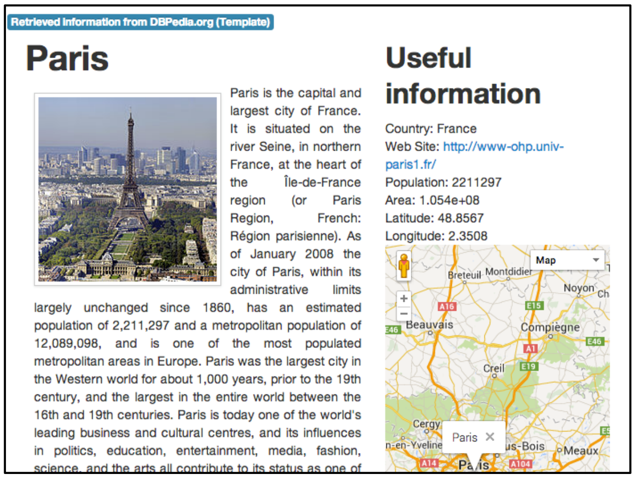
```
<span about={{_id}} typeof="City">
<h1 property="name">{{city}}</h1>
…
</span>
```

In the code above we defined an object with identification "{{_id}}" and that has an RDF type "City", from the vocabulary http://schema.org/City. This vocabulary states that a City has a property "name", and we will give it the value "{{city}}". This syntax comes from the templating engine we use (mustache.js).

The JavaScript code below fills the template with real values:

```
var pagedata = {};
pagedata.city = city; // dbpedia result
article = wikinextHelper.city(pagedata);
```

[17] http://wikinext.gexsoft.com/wiki/519e04c580194c4178000001
[18] http://dbpedia.org/



Retrieved information from DBPedia.org (Template)

Paris

Paris is the capital and largest city of France. It is situated on the river Seine, in northern France, at the heart of the Île-de-France region (or Paris Region, French: Région parisienne). As of January 2008 the city of Paris, within its administrative limits largely unchanged since 1860, has an estimated population of 2,211,297 and a metropolitan population of 12,089,098, and is one of the most populated metropolitan areas in Europe. Paris was the largest city in the Western world for about 1,000 years, prior to the 19th century, and the largest in the entire world between the 16th and 19th centuries. Paris is today one of the world's leading business and cultural centres, and its influences in politics, education, entertainment, media, fashion, science, and the arts all contribute to its status as one of

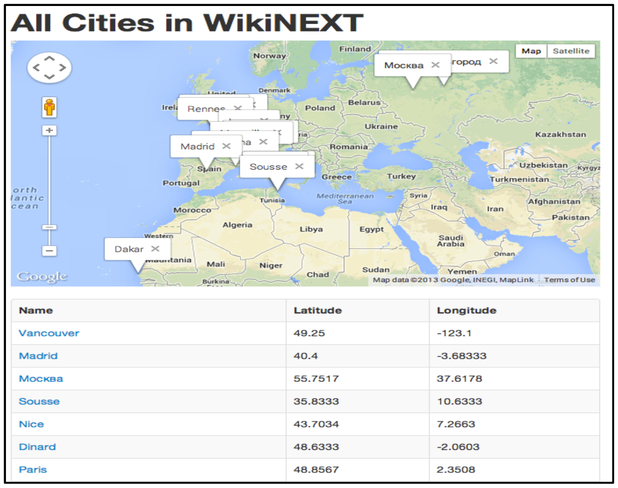
Useful information

Country: France
Web Site: http://www-ohp.univ-paris1.fr/
Population: 2211297
Area: 1.054e+08
Latitude: 48.8567
Longitude: 2.3508

Figure 5: Results using a template.

4.3 Reusing data

To demonstrate data reuse, we try to recombine data sources on the wiki, creating what is typically known as mash-up by developing a web application that combines all the created cities on WikiNEXT. The **Error! Reference source not found.** displays the output map of the wiki application that retrieves the location of all cities available in our knowledge base and shows all cities on a map and in a table. WikiNEXT uses SPARQL to query its knowledge base. Hence, queries can be written without having to deal with semantic heterogeneity problems since we use predefined ontologies.



All Cities in WikiNEXT

Name	Latitude	Longitude
Vancouver	49.25	-123.1
Madrid	40.4	-3.68333
Москва	55.7517	37.6178
Sousse	35.8333	10.6333
Nice	43.7034	7.2663
Dinard	48.6333	-2.0603
Paris	48.8567	2.3508

Figure 6: Map view of all cities in WikiNEXT

5. Discussion and Conclusions

WikiNEXT is a “programmable wiki for the Linked Data”: a wiki in which we can write documents with an application layer that exploits the web of Data. Applications are written using standard HTML/CSS/JavaScript, and can be shared, cloned, improved collaboratively. We also designed an API and protocols for keeping a tight, synchronous link between the data embedded in the pages and the databases on the server side. The demonstration we propose consists in presenting live the WikiNEXT principles for writing documents and linked data applications directly in the browser. We propose to reproduce live, the City scenario described in the paper, showing how WikiNEXT simplifies the programming of such applications, compared to classical solutions that involve Desktop Based tools. WikiNEXT is used now at the University of Nice/France and by the w3devcampus, the online platform that hosts the HTML5 W3C course.

6. REFERENCES

[1] Krötzsch, M., Vrandečić, D., & Völkel, M. (2006). Semantic mediawiki. In The Semantic Web-ISWC 2006 (pp. 935-942). Springer Berlin Heidelberg.

[2] Schaffert, S. (2006, June). IkeWiki: A semantic wiki for collaborative knowledge management. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on (pp. 388-396). IEEE.

[3] Buffa, M. Gandon, F., Ereteo, G. Sander, P. & Faron, C., SweetWiki: A semantic wiki, Special Issue of the Journal of Web Semantics on Semantic Web and Web 2.0, Volume 6, Issue 1, February 2008 , Edited by Mark Greaves and Peter Mika, Elsevier, Pages 84-97

[4] Auer, S., Dietzold, S., & Riechert, T. (2006). OntoWiki–A tool for social, semantic collaboration. In The Semantic Web-ISWC 2006 (pp. 736-749). Springer Berlin Heidelberg.

[5] Buffa, M. “*Intranet Wikis*”, workshop “intrawebs” of WWW 2006, Edinburgh, Scotland

[6] Krotzsch, M., SCHAFFERT, S, & Vrandecic, D. Reasoning in semantic wikis. In: Reasoning Web. Springer Berlin Heidelberg, 2007. p. 310-329.

[7] Oren, E., Breslin, J. G., & Decker, S. (2006, May). How semantics make better wikis. In Proceedings of the 15th international conference on World Wide Web (pp. 1071-1072). ACM

[8] Oren, E., SemperWiki: a semantic personal Wiki. In SemDesk. 2005.

[9] Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., ... & Lee, R. (2009). Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In *The semantic web: research and applications* (pp. 723-737). Springer Berlin Heidelberg.

[10] Schaffert, S., Eder, J. Samwald, M., & Blumauer. A., Kiwi - knowledge in a wiki. In European Semantic Web Conference 2008, 2008

[11] Buffa, M., Delaforge, N., Erétéo, G., Gandon, F., Giboin, A., & Limpens, F. (2013). ISICIL: Semantics and Social Networks for Business Intelligence. In SOFSEM 2013: Theory and Practice of Computer Science (pp. 67-85). Springer Berlin Heidelberg.

[12] Di Iorio, A., Musetti, A., Peroni, S., Vitali, F.: Ontology-driven generation of wiki content and interfaces. The New Review of Hypermedia and Multimedia 16(1&2): 9-31 (2010)

[13] Rospocher, M., Ghidini, C., Pammer, V., Serafini, L., Lindstaedt, S.: MoKi: the Modelling wiKi. SemWiki 2009

[14] Hernández A.G and García. M. N. M., A JavaScript RDF store and application library for linked data client applications. In Proceedings of the 21st International World Wide Web Conference, WWW2012, 2012.

[15] Meilendera, T., Jayb, N., Lieberb, J., & Palomaresa, F. Semantic wiki engines: a state of the art. Semantic-web-journal.net. IOS Press, 2010.