A Wiki Way of Programming for the Web of Data

Pavel Arapov
WIMMICS team
I3S Laboratory
arapov@i3s.unice.fr

Michel Buffa
WIMMICS team
I3S Laboratory
buffa@i3s.unice.fr

Amel Ben Othmane
WIMMICS team
INRIA
amel.ben-othmane@inria.fr

ABSTRACT
WikiNEXT is a wiki engine that enables users to write rapidly applications directly from the browser, in particular applications that can exploit the web of data. WikiNEXT relies on semantic web formalisms and technologies (RDF/RDFa lite) to describe wiki page content and embedded metadata, and to manipulate them (for example, using the SPARQL language). WikiNEXT is a mix between a web-based IDE (Integrated Development Environment) and a semantic wiki. It embeds several editors (a WYSIWYG editor, and an HTML/JavaScript editor + a JavaScript library manager) for coding in the browser, provides an API for exploiting semantic metadata, and uses a graph based data store and an object oriented database for persistence on the server side. It has been specially designed for writing online programming tutorials (i.e. an HTML5 tutorial, a semantic web tutorial on how to consume linked data, etc.), or more generally for developing web applications that can be mixed with more classical wiki documents (in fact all WikiNEXT pages are web applications). 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.

General Terms
Algorithms, Performance, Design, Experimentation, Languages.

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

1. INTRODUCTION
The evolution of the Internet from static pages to web applications changed the paradigm of web programming. Furthermore, technologies and trends changed at a very fast pace these two last years, with the rise of HTML5 and the appearance of powerful JavaScript frameworks, making web-based IDEs (Integrated Development Environments) possible, such as jsbin.com, jsfiddle.com or c9.io. These online applications make possible to write and test JavaScript code directly into the browser. Their online status makes easy to share your code or clone someone else's code. With WikiNEXT, we tried to mix the writing of web application proposed by web based IDEs, with features proposed by semantic wikis: formal representations of data in a graph based knowledge base, reasoning, requesting the web of data, annotation of documents, etc. We will show in this paper the dynamic creation of data models for the knowledge base and the manipulation of these data by web applications written directly using the wiki online editors.

2. STATE OF THE ART
Web-Based IDEs: JavaScript based source code editors like Code Mirror[3], ACE Cloud9[4] or ternjs[5] 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.

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) or OWL (Web Ontology Language). 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[6] is often used internally for querying RDF data. Today, semantic wikis are still active and there are new solutions, such as the Halo[7] extension of Semantic Media Wiki, that proposes forms, auto-completion, a WYSIWYG editor; or more complex extensions such as MoKi [8] (focuses on enterprise modeling) and OWiki [7] (ontology-driven generation of wiki content), also based on Semantic Media Wiki. The KiWi project (Knowledge in Wiki [5]), successor of IkeWiki, or SweetDeki [6] the wiki that succeeded to SweetWiki[3], and has been integrated in the ISICIL ANR project [6] are more like Content Management Systems with semantic enhancements. In the next section we present WikiNEXT, which is a mix between a semantic wiki and a Web-based IDE.

3. WikiNEXT
Enterprise wikis, such as MindTouch, Confluence, Sharepoint, powerful solutions for knowledge management, can be considered more as content management systems (CMS) that provide different modules: wiki, blogs, social network, etc. These modules

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

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577349

[3] http://codemirror.net/
[4] http://ace.c9.io/
[5] http://ternjs.net/
[6] http://www.w3.org/TR/sparql11-query/
[7] http://www.projecthalo.com/

are very often developed as an extension or as a plugin and often require server side development, as well as defining a data model. WikiNEXT makes this possible using only client side development, directly in the browser.

A WikiNEXT page has two parts and two editors consensually: HTML and JavaScript. The HTML part holds the view part and the metadata. We think that it fits well the MVC pattern: where Model – metadata, View – HTML and Controller – JavaScript code of a wiki page. See Figure 1.

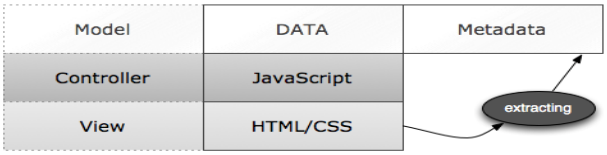

Figure 1 MVC pattern of a wiki page

Each page is represented semantically as a named graph, whose name is based on its URI (Uniform Resource Identifiers), see Figure 2. We call this graph the *Local Knowledge Base*. The full stack of wiki pages give us the *Global Knowledge Base*, the union of all the named graphs.

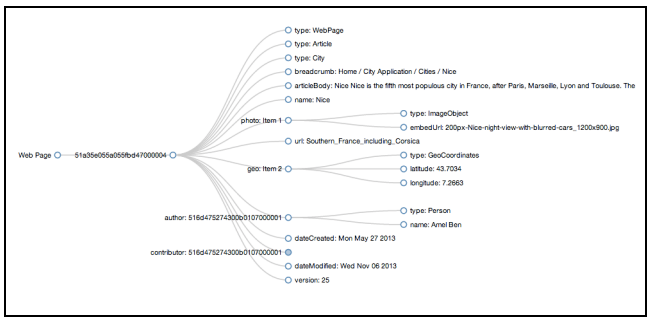

Figure 2 Semantic Graph of a wiki page

WikiNEXT pages are associated with a set of metadata that describes their basic characteristics: Title, Article, Author, Contributor, Date, etc. But pages are also containers: they can hold metadata created manually or added automatically by an application. WikiNEXT allows defining the data model as a template in a wiki page. This page could then be reused by wiki applications. The template is based on HTML syntax with RDFa/RDFa lite[8] annotations and uses the Mustache[9] logic-less template engine. WikiNEXT uses schema.org[10] vocabularies to describe the default structure of articles, users' details, etc.

Implementation

WikiNEXT has been written from scratch in JavaScript and relies on the Node.js HTTP server. Persistence uses two databases: a MongoDB database for storing JavaScript objects and RDFstore-js for storing semantic metadata and SPARQL requesting. Node.js[11] (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 triplets are also persisted in MongoDB[12], then, RDF triplets are both accessible through RDFstore-js using SPARQL, but also directly through MongoDB for programmatic use. 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. Evaluation and conclusion

WikiNEXT proposes a new approach to programming the semantic web in JavaScript, by mixing a web-based IDE and a semantic wiki. It's currently used at the University of Nice. We wrote some online tutorials, using WikiNEXT itself, and set up a testing protocol[13]. We choose a group of people with a diverse level of web programming knowledge, and asked them to complete various tutorials available in WikiNEXT, that go from writing very simple examples to developing more complex applications. We measured the time spent for completing each tutorial, observed behaviors of our testers and took note of all the problems encountered. Finally we conducted an interview, and asked them to compare this experience with classical web development techniques for performing the same tasks (e.g. using Java/Eclipse). Results[14] showed that the system is easy to use, saves time and enables easy data reuse between applications, in particular writing HTML/CSS/JavaScript code, using the WikiNEXT API for creating pages, talking to external or internal data sources was very simple.

5. 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, Gandon, Ereteo, Sander, & Faron, 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] Sebastian Schaffert, Julia Eder, Matthias Samwald, & Andreas Blumauer. Kiwi - knowledge in a wiki. In European Semantic Web Conference 2008, 2008

[6] 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.

[7] Angelo Di Iorio, Alberto Musetti, Silvio Peroni, Fabio Vitali: Ontology-driven generation of wiki content and interfaces. The New Review of Hypermedia and Multimedia 16(1&2): 9-31 (2010)

[8] Marco Rospocher, Chiara Ghidini, Viktoria Pammer, Luciano Serafini, Stefanie N. Lindstaedt: MoKi: the Modelling wiKi. SemWiki 2009

[8] http://www.w3.org/TR/xhtml-rdfa-primer/, http://www.w3.org/TR/rdfa-lite/

[9] http://mustache.github.io/

[10] http://schema.org/, http://schema-rdfs.org have been created by the main search engine vendors and propose several RDF/S schemas.

[11] http://nodejs.org

[12] http://www.mongodb.org

[13] http://wikinext.gexsoft.com/wiki/52304bb57a61be9c29000010

[14] http://wikinext.gexsoft.com/wiki/52304d287a61be9c29000011