RZ 3654 (# 99664) 04/04/06
Computer Science 5 pages

Research Report

First Explorations towards Semantic Wiki

Daniela Bourges-Waldegg, Marcel Graf, Christian Hoertnagl

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{dbw, grm, hoe}@zurich.ibm.com

Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

First Explorations towards Semantic Wiki

Daniela Bourges-Waldegg, Marcel Graf, Christian Hoertnagl

IBM Zurich Research Laboratory, Saeumerstrasse 4
CH-8803 Rueschlikon, Switzerland
{dbw, grm, hoe}@zurich.ibm.com

Abstract. This position paper explores the novel design space for semantic wikis at the intersection of semantic markup and collaborative website editing. It explains how semantic wikis can help in bootstrapping the Semantic Web by injecting initial information, provided that they keep the simple user experience from conventional state-of-the-art wiki systems intact. It sketches a semantic wiki syntax M3, which accounts for three major design criteria: (1) ease-of-use comparable to conventional wiki syntax; (2) full semantic expressiveness on the same level as canonical RDF notations; (3) extensibility for accommodating important ontologies with pluggable shorthand notations. The paper concludes with a conceptual design for a semantic wiki as extension to an existing open source wiki engine.

1 Introduction

Wikis constitute a popular type of Web 2.0 [1] infrastructure that allows collaborative web site editing. They employ different varieties of simple wiki syntax to convey the structure of content in terms of its font formatting, separation into bullet lists, etc. This has led to the creation of a new type of content by communities of many authors, with Wikipedia, a free encyclopedia of currently about one million articles in the English version, as the premier cited example.

While wikis are highly popular, the Semantic Web [2] so far has been entangled in a chicken-and-egg problem, whereby it is perceived that content owners have to replace or augment their content and HTML markup by manual exercise; yet they have little motivation to do so before semantic browsers take off, while at the same time new browser development is hampered by the lack of significant content.

We see semantic wikis that can capture both structural and the semantic properties of content as one potentially important tool to help bootstrap the Semantic Web, because they allow wiki content owners to opt-in and add semantic information incrementally (or otherwise to stay with their current practice as-is). In addition, we wish to use a semantic wiki for maintaining Resource Description Framework (RDF) models in use by several of our other research prototypes.

Extrapolating from Web 2.0 phenomena in general and from wikis in particular, it seems to us that the key success factor is to keep the simple user experience that is now associated with wikis intact as best as we can while adding new expressivity at the same time: an author who does not want to provide semantic information, should be allowed to do so, and should not be bothered with any new or modified syntax. What we have in mind is an extension to wiki syntax that ultimately allows capturing of any RDF statements and that can be seen as a straightforward and faithful extension to existing practice.

So far we have identified three initial design criteria for this add-on: (1) ease-of-use comparable to conventional wiki syntax; (2) full semantic expressiveness on the same level as canonical RDF; (3) extensibility for accommodating important ontologies with pluggable shorthand notations.

We would like to access the information stored in a semantic wiki system in at least the following two ways: (1) query the aggregated RDF model for use in other applications; (2) extend the wiki browser front-end into a semantic browser. This is further outlined in Section 3 and also shown in Figure 1.

2 M3 notation

We have chosen a particular wiki syntax, Markdown [3], as a basis for our extensions; however, they do not specifically depend on this dialect but could be encapsulated inside other choices as well. Given that Markdown (with an M) covers syntax, and that we take some inspirations for proposed extensions from Notation 3 [4] (or N3), we'll refer to the resulting semantic wiki syntax as M3. This name also summarizes two of our design criteria: ideally M3 should be as simple and expressive in the semantic wiki domain, as N3 is in the semantic model (RDF) domain.

We have by now designed a preliminary BNF grammar for M3 and are in the process of validating it with formal parser generators and other tools to arrive at a first executable prototype. The grammar already covers our own examples for constructing medium-sized RDF graphs, as well a standard examples from the World Wide Web consortium (W3C; e.g. in [5]). Having said this, our proposed extensions and requirements may change as we gain more practical experience in this space. Within this short position paper, we'll confine ourselves to the following brief example and an informal explanation of the few M3 constructs it employs (these are highlighted below in bold font). The example concerns two quotes, plus semantic statements referring to the dictionary definition of some contained words (from an RDF representation of WordNet [6]) as well as their author (from the Dublin Core metadata element set [7]).

```
{{@prefix     WordNetHandler}}
{{@prefix dc: http://purl.org/dc/elements/1.1/
              DublinCoreHandler}}
Some quotes
===========
((
I wonder why I wonder why.
I wonder why I wonder.
I wonder _why_ I wonder why I wonder why I wonder.
dc:""Richard P. Feynman"" << dc:creator (1918 - 1988)
))
Good ""resolutions"" are like babies crying in church,
they should be ""carried out"" immediately.
```

The preliminary M3 language design respects the following concepts in order: (1) Data comes before metadata in the raw textflow, and there is clear syntax to separate the two. Within the scope of this paper we refer to what is rendered inside a wiki page as data (e.g. ”resolutions”), and to further descriptions of data that will be excluded from the rendered textflow as metadata (e.g. dc:creator). (2) The syntactic design space for M3 as a (largely) backward-compatible extension to conventional wiki dialects is limited by the fact that these dialects reserve many special characters, such as brackets, for their own purposes. Partly for this reason, but also to reserve a distinct kind of character tokens for metadata (which is as well distinct from the data that it describes) all metadata is marked by double characters such as “>>” or “<<”. Ultimately, this choice should allow M3 authors to easily associate double characters with metadata, given that it is simple to think of the latter as “data about data”. As a case in point, all first terms immediately following “<<” are metadata (e.g. dc:creator). (3) As does N3, we use RDF statements (triplets of subjects, predicate, and objects [2], or *S*, *P*, and *O*) as the principal semantic building blocks that M3 encodes. By the nature of the Semantic Web’s core technologies, RDF statements form edges in the directed labeled graph that ultimately represents every RDF model, and we use the “>>” and “<<” tokens to indicate the direction from the origin (subject) to the tip (object) of RDF statements. In other words, the syntax allows for both *S* >> *P* >> *O* and *O* << *P* << *S* constructions (given that we want to consistently place data before metadata). If one of the three elements is missing, it can usually be inferred. For instance, the construct dc:""Richard P. Feynman"" << dc:creator encodes that the first quote stems from Richard P. Feynman: the subject of this statement is resolved to the surrounding citation. (In N3 notation, this corresponds roughly to the statement “:quote1 dc:creator :feynman”, plus possibly further statements in relation to the object.) The resolver looks for a surrounding M3 closure (as marked by the “((“ delimiters) or otherwise automatically refers to structures implicit in the normal wiki syntax, or to the wiki page as a whole. This approach permits the additional abbreviated constructions *S* >> *P* and *O* << *P*.

M3 offers a variety of ways to specify RDF resources and their URIs, such as references to existing ontologies (e.g. in dc:creator), M3 closures, references to wiki pages, and normal hyperlinks. One versatile way concerns text in double-double quotes (e.g. dc:””Richard P. Feynman””): such text is passed into pluggable RDF encoders that may derive arbitrarily many statements and return (the URI of) one RDF node. RDF encoders are executables (e.g. a DublinCoreHandler Java class in the context of a Java-based wiki engine) that can be configured by attaching them to prefixes, similarly to established practice in relation to namespace URIs. (The concept of a default RDF encoder, WordNetHandler in the example, is also similar to the namespaces practice.) By convention, RDF encoders translate ontology-optimized convenience notation into RDF models in the namespace of a given ontology, such as the Dublin Core ontology (e.g. deconstruct “Richard P. Feynman” into first name and surname, or translate “resolutions” into singular form that can be looked up in WordNet dictionary). Other instances of convenience notations allow domain-specific mappings; for instance, in life sciences a convenience notation may e.g. refer to or expand into entire semantic models of complex proteins (as are already in use in this domain) starting from a simple symbolic id notation.

RDF encoders are registered inside double-curly brackets “{{“; these delimiters also distinguish other advanced M3 features. For instance there is support for a set of pragmas other than “@prefix”, such as for defining macros, or for including serializations or visualizations of the produced RDF model. Furthermore, raw RDF statements can be cited here in a mode very similar to plain N3. Our M3 wiki engine will offer a non-expert mode, where these advanced constructs (inside the “{{“ delimiters) are not allowed (but where some global RDF encoders may still be pre-set as part of the M3 wiki configuration). In effect, M3 constructs can be separated into two layers, simple and advanced, and in combination they should fulfill the desired design criteria, namely simplicity of use especially for common cases, and if needed also full expressivity for building any type of RDF model from inside wiki pages.

3 First prototype design

When it comes to implementing an M3 semantic wiki system on top of an existing wiki engine at least three control points are required: (1) The wiki engine has to be instructed to persist its pages in M3 notation rather than its native wiki syntax. (2) The wiki engine has to offer pages for editing in M3 notation likewise. (3) The wiki engine has to be able to render M3 syntax into normal (X)HTML (possibly plus semantic annotations) that can be rendered for users who visit and view wiki pages.

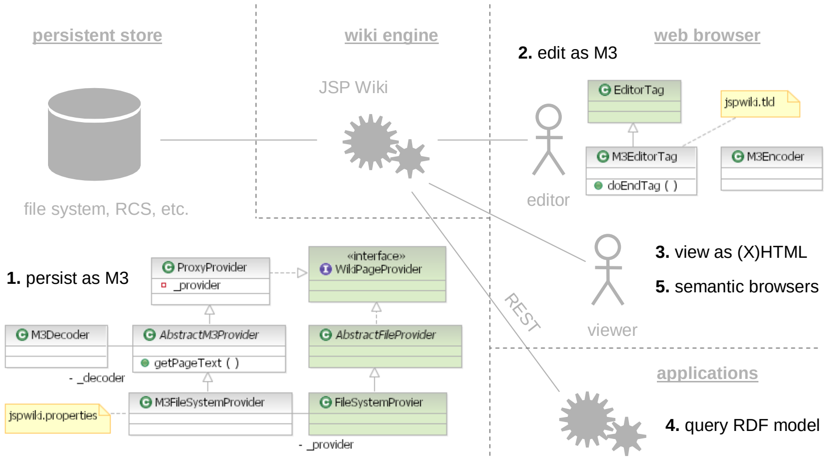

Fig. 1. First prototype design for configuring M3 wiki on top of the JSP wiki engine.

We have for now chosen JSP Wiki [8] as a basis of our prototype, and as Figure 1 indicates, we have identified all required control points in its implementation in a way that allows us to introduce the necessary modifications by simply registering additional Java classes in two JSP Wiki configuration files. Further details about the emerging prototype implementation are beyond the scope of this position paper and will be reported separately. One thing to point out is that JSP Wiki’s native wiki syntax is distinct from Markdown, hence working with both this wiki engine and that wiki syntax forced us to aim at a fairly flexible (and hence potentially portable) design from the very beginning.

Figure 1 also indicates two ways in which semantic information from an M3 wiki can be utilized: (4) query of the RDF model resulting from all (or some) pages e.g. via a Representational State Transfer (REST) API; (5) semantic browser as extension to normal wiki front-end. In the chosen example domain (literary quotes), the latter could e.g. give quick access (by right-clicking on a word, say) to quotes dealing with synonyms or contraries of that word, thus making convenient use of the semantic model for WordNet. We have also started work on such browsers and are considering specific use cases in the areas of life sciences and configuration management.

4 Related work and conclusions

M3 syntax embeds semantic information inside normal wiki syntax, in a similar way as RDF/A [5] and the RDF/XML serializations embed it inside (X)HTML or other XML markup.

We are particularly interested in the potential of semantic wikis for bootstrapping the Semantic Web. Hence, in partial contrast to other emerging semantic wiki approaches [9] our aim is first to provide simple access to existing ontologies and then also to allow a bottom-up creation of new ontologies. The usage of handlers results in such flexibility. These handlers, as well as namespaces, allow authors to refer to resources that are already defined outside of the wiki, as well as on wiki pages; in the latter case, our proposed syntax envisages the definition of resources at a granularity finer than that of a whole wiki page. On balance we deliberately do not define a fixed set of ontology-specific terms as part of the M3 core language, but instead provide an extension mechanism for accommodating any (also future) ontology in equal terms.

References

1. O’Reilly, T.: What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software (September 2005). http://www.oreillynet.com/lpt/a/6228
2. Berners-Lee, T. Hendler, J., Lassila, O.: The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American (May 2001).
3. Gruber, J.: Markdown: Syntax. http://daringfireball.net/projects/markdown/syntax
4. Berners-Lee, T.: Notation 3: An RDF language for the Semantic Web (1998–2001). http://www.w3.org/DesignIssues/Notation3.html
5. Adida, B., Birbeck, M.: RDF/A Primer 1.0: Embedding RDF in XHTML. World Wide Web consortium (W3C), draft 2006-01-24.
6. Fellbaum C. (ed): WordNet: An Electronic Lexical Database, MIT Press (1998).
7. DCMI Usage Board: CMI Metadata Terms, DCMI Recommendation (2005-06-13).
8. JSP Wiki documentation. http://jspwiki.org/wiki/JSPWikiDocumentation
9. Völkel, M et al.: Semantic Wiki State of the Art, paper in progress, 2005–2006. http://wiki.ontoworld.org/index.php/Semantic_Wiki_State_of_The_Art_Paper