Chapter XXXIV

Wiki Semantics via Wiki Templating

Angelo Di Iorio
Department of Computer Science, University of Bologna, Italy

Fabio Vitali
Department of Computer Science, University of Bologna, Italy

Stefano Zacchiroli
Universitè Paris Diderot, PPS, UMR 7126, Paris, France

ABSTRACT

A foreseeable incarnation of Web 3.0 could inherit machine understandability from the Semantic Web, and collaborative editing from Web 2.0 applications. We review the research and development trends which are getting today Web nearer to such an incarnation. We present *semantic wikis*, *microformats*, and the so-called "lowercase semantic web": they are the main approaches at closing the technological gap between content authors and Semantic Web technologies.

We discuss a too often neglected aspect of the associated technologies, namely how much they adhere to the wiki philosophy of open editing: is there an intrinsic incompatibility between semantic rich content and unconstrained editing? We argue that the answer to this question can be "no", provided that a few yet relevant shortcomings of current Web technologies will be fixed soon.

INTRODUCTION

Web 3.0 can turn out to be many things, it is hard to state what will be the most relevant while still debating on what Web 2.0 [O'Reilly (2007)] has been. We postulate that a large slice of Web 3.0 will be about the synergies between Web 2.0 and the Semantic Web [Berners-Lee et al. (2001)], synergies that only very recently have begun to be discovered and exploited.

We base our foresight on the observation that Web 2.0 and the Semantic Web are converging to a common point in their initially split evolution lines. On the Web 2.0 side, even if specifications of its precise nature are still lacking, it is settled that Web 2.0 has changed many aspects of the plain old Web, and has done so pivoting around the concept of *collaboration* [O'Reilly (2007); Musser & O'Reilly (2006)]:

technically collaboration has been made easier by a new approach at web application development (AJAX) which has leveraged the potentialities of web applications and improved user experiences, still requiring only a web browser to participate;

socially the advent of social networking sites has enabled millions of users to find each other and chime in via affinities in interests;

economically a new business model—based on exploiting user-provided content and using added value services in convincing them to provide more (the more the content, the better the service)—

has closed the circle attracting big companies in the game.

In spite of Web 2.0 turning into a reality in just a couple of years, the Semantic Web [Berners-Lee et al. (2001)] envisaged by Tim Berners-Lee since the late nineties[1] is, in the eyes of many web users, still a blurry, non implemented concept. The reasons for this acceptance delay are, by comparison with the history of Web 2.0, mostly to be found in a chicken and egg scenario. Users are not encouraged to provide semantically rich content since added value services for such kind of content are missing; companies are not seeing the potential market since there are no users.

Things are made worst by the "height" of the Semantic Web technology stack: there are too many technologies to master for adding semantic annotations to personal home pages or blog posts. *Authors*, the key figures which made Web 2.0 a success, are kept out of the Semantic Web loop as they do not have the capabilities to master the needed technologies (RDF [Manola & Miller (2004)], OWL [McGuinness & van Harmelen (2004)], SPARQL [Prud'hommeaux & Seaborne (2008)], to mention just a few); this is a key difference with the simplicity authors are used to with wiki and blog engines.

Recent trends [Hendler (2008)] seem to be showing a way out: semantically rich data sets coming from governments and research projects are being published; a handful of start up companies have started businesses exploiting Semantic Web technologies in particular domains; even the standardization tracks of Semantic Web-related languages have shown an acceleration in the past 2 to 3 years. But such advancements are far from bringing Semantic Web to the masses as all of them are relegated to scientific or corporate niches. More importantly, they still fail to address the authorship problem, as they are usually not interested in closing the gap between authors and Semantic Web technologies.

To back our initial claim, we observe that two yet to be mentioned recent trends are diminishing the distance between authors and Semantic Web technologies; interestingly enough they are doing so in two key environments of Web 2.0: wikis and blogs. The first trend is that of *semantic wikis* which are bringing semantic annotation capabilities to authors, yet requiring no more knowledge than that needed to contribute to Wikipedia. The second trend is that of *microformats* and, more generally, of the "lowercase semantic web". Microformats are empowering users of simplified content management systems, such as blog engines, to add semantic annotations exploiting capabilities readily present in the legacy languages, e.g. XHTML, already used by authors. Using microformats authors gain immediate benefits—such as fancy CSS-based layouts—not necessarily related to the machine understandability of the (now) annotated content.

The aim of this chapter is to introduce the reader to the research and technological trends related to semantics wikis and the lowercase semantic web. We will see where they generated from, what they are heading to, and how they can be used to produce semantically rich content which is ready to be consumed not only by forthcoming implementations of Semantic Web technologies, but also by readily available, though implementation-specific, added value services.

Of the two trends, the chapter will then focus on the wiki side, critically reviewing some of the design principles of mainstream semantics wikis. The main objection to most of them is that they are neglecting the free editing philosophy Leuf & Cunningham (2001) which brought wikis to success. The issues however can be easily fixed by putting established research results into use. Our proposed tentative solution is to reuse wiki templating mechanism, piggy backing on them the desired semantic information; the advantage of such a solution is to avoid drifting from an authoring practice (templating), which is already part of wiki author work-flow. Finally, the reader will be pointed to the needed resources for pursuing a similar critical review exercise on the blog side for what concerns the microformat trend.

BACKGROUND

1the first working draft of the once called "Resource Description Framework (RDF) Model and Syntax" specification was published in August 1997, see the revision history of http://www.w3.org/TR/WD-rdf-syntax-971002/

This section provides background for the two trends we will be discussing: microformats and semantic wikis. The two sets of apparently unrelated technologies share the objective of easing the access to "semantic authorship". This means that they both provide lightweight mechanisms for enriching the plain content which can be found on the web with semantic information, so that it can be later on automatically processed by a computer program.

Traditionally, microformats were born in the blogosphere [Barlow (2007)] (the user and developer communities related to weblogs) while semantic wikis are an evolutionary trend of wikis. The pedigree difference had an impact on the respective trends: while microformats exploits details of XHTML (which directly or indirectly is the language most frequently typed by blog post authors), semantic wikis provide extended wiki markups or interfaces to specify semantic annotations.

Chapter structure: The next section discusses microformats as an important contribution to ease semantic authorship, while the remainder of the chapter will focus on semantic wikis contributing criticisms and proposing novel solutions. For the interested reader, the proposal is to try parallel that work on microformats and blogs.

Microformats and the "lowercase semantic web"

Consider the following scenario:

> *You have been visiting an abroad workplace for the past 6 months. You have been blogging about that regularly both for your old friends at home (to keep them informed), and for your new friends abroad (to let them know you better).*
> *6 months have passed; it's time to leave and you are organizing a* goodbye party. *You want to blog about the event details and enable your local friends to add the event to their calendars as easy as possible.*

The above scenario is a simple example of what we call "everyday semantic web". In such a scenario we are not requiring a full fledged implementation of Semantic Web [Berners-Lee et al. (2001)] where intelligent agents take care to schedule appointments depending on several factors (proximity, availability, agenda matching, …). Rather, we just want to use the web as a media to convey semantic information (an appointment in this case), fulfilling a simple requirement: doing so using a Web application which is only known to support some widespread markup language (e.g. the application being our blog engine, the language (X)HTML).

Implementing the scenario following the Semantic Web way would lead us to use RDF Calendar [Connolly & Miller (2005)]: after typing in the blog engine the post prose, we would need to encode the event information in RDF Calendar, upload it separately (admitting that our blog engine permits that, otherwise we would need to embed the information), and link the RDF file from the blog post. Alternatively, we can go for the vCalendar standard [Alden & Bartlett (1996)], and create a `.vcs` file containing just the event information, upload and link it from the blog post. Our friends can then click on the `.vcs` file and import it in their agenda application. In both cases we are requiring the author to code the information twice: one for humans, one for the machine. Considering that the author is probably blogging in her spare time, she will not be particularly happy about the required extra burden: that is why we have all been diligently copying and pasting the information by hand in our agenda application thus far.

The hCalendar[2] microformat comes to the rescue. Using it we can encode the semantic information side by side with the prose and only once, by using cleverly (X)HTML classes. Here is an example:

```
<div class="vevent">
 <span class="summary">Goodbye party</span>
 <p>Finally, I'm leaving :-( I would welcome all of you for the
```

2http://microformats.org/wiki/hcalendar

```
  due goodbye party. Let's meet <span class="location">in front
  of the lab</span> <abbr class="dtstart"
   title="2008-10-15T13:30:00+02:00">today at 13:30</abbr>, I'll
  be around until <abbr class="dtend"
  title="2008-04-15:00:00+02:00">15:00</abbr>, as later on I'll
  have to catch a plane :-)
 </p>
</div>
```

From the author point of view this representation is convenient for various reasons. First of all information are encoded (mostly) once.[3] This diminish the likelihood of information getting out of date upon updates, is less likely to hinder author willingness to add semantic annotation (and hence has also good chances to help technology diffusion), and can work around technical limitations of the web application in use: the above markup only requires the ability to specify the `class` attribute on content parts, such a requirement is fulfilled by many (simplified, as blogs, or full-fledged) content management systems even when (X)HTML is not the language directly used by authors for typesetting content.

A second advantage for authors is that microformats implementations exist out of the box: they can serve the author only relying on an agreement upon class names. At the very minimum, authors using hCalendar class names can found on the web a plethora of CSS stylesheets which can be plugged in their blogs to have event details stand off from the ordinary text, distinguishing them visually from the plain prose. This aspect is shared by several microformats and implements the principle of *instant gratification* for authors: by simply choosing appropriate names, authors obtain fancy renderings, while (knowingly or not) having just provided a new bit of semantic information on the Web.

From the point of view of content users, the life is not (yet) completely trivial, but it is changing very rapidly. Indeed for the random reader of the blog post the event will just be fancy, but is far from being just one click away from its agenda application. Still, it is not *that* far. For instance, just installing the Operator[4] extension for the Firefox browser, users will be notified when a page contains microformat instances. In the particular case of hCalendar, Operator offers a contextual menu enabling direct importation in an agenda application, via vCalendar conversion.

Microformats [Khare (2006)] do not stop at hCalendar of course, there is a full set of microformats to support various needs. The shared principles are however the same. The key idea is to exploit the capabilities of the language used by content authors, in particular those of XHTML. In a spectrum ranging from the simplest to the most complex microformat, we can distinguish between elemental and compound microformats. *Elemental* (or “link-based”) microformats simply revamp a traditional principle of hypertexts [Bieber et al. (1997)], neglected in modern implementations: *link classification*. By augmenting the set of allowed values for the `rel` attribute of anchor elements (a), elemental microformats can express properties of the source page of the link. Several examples of elemental microformats are in order:

the license of a page content can be specified using the `rel-license` microformat: adding

```
<a href="http://creativecommons.org/licenses/by-sa/3.0/"
 rel="license">...</a>
```

to a page states that its content is licensed under the terms of the Creative Commons BY-SA license version 3.0

the tags associated to a blog post, or to some page in a CMS, can be specified using the rel-tag microformat: it is similar to rel-tag, but uses the `rel="tag"` relationship

3“mostly” is embodied by the `title` attributes used to input machine parseable dates and times. This is a technical limitation of current implementations: libraries for parsing dates out of English texts (e.g. “today at 2pm”) have been available to Perl hackers for several years

4https://addons.mozilla.org/it/firefox/addon/4106

the human relationship among the owner of a homepage and the owner of a linked homepage can be specified using the XFN (XHTML Friends Network) microformat. With XFN you can describe relationships such as: friendship degree (contact, acquaintance, friend), romantic involvement (crush, dated, sweetheart), identity (myself), …

no relation among two linked pages can be explicitly required using the rel-nofollow microformat, which has been designed to avoid influencing search engines, for example to fight abuse of blog post comments by spammers

Compound microformats on the other hand have been designed to encode relatively complex information such as calendar entries. Though hCalendar is probably the most popular compound microformat, other examples of compound microformats are hCard (to encode business card information sets, which are ubiquitous on the web thanks to homepages) and hReview (to represent reviews and rating of entities represented on the web). A full directory listing of microformat specifications, both elemental and compound, is available on the `microformats.org` wiki[5].

But what does it mean for a microformat to have a specification? All mentioned microformats rely on only three aspects of XHTML: the multi-valued attributed `class`, new allowed values for `rel`, and the intrinsic nesting of XHTML (exploited for example by hCalendar: the details of the described event have to be found inside the element annotated with `class="vevent"`). Hence a microformat specification can be very simple, it only takes a description of prescribed `class` names and `rel` attribute values; such information can be described in an uniform way using the XMDP (XHTML Metadata Profile) format. The description of the meaning of names and values is described as informally as in a W3C specification.

The good thing about specifications is usually that they foster implementations. This, together with the simplicity of the average microformat specification, has indeed worked also for microformats and their implementations are spreading. Some notable examples are: `rel-license` is implemented by the Creative Commons license chooser[6], `rel-tag` by popular blog engines, `rel-nofollow` by Google spiders, compound microformats sports several tool able to "parse" them into their corresponding legacy formats (e.g. Operator/hCalendar/vCalendar). The future of implementations is very encouraging: current implementations are listed on the microformats wiki, and Firefox 3.0 comes with a built-in API[7] to work on microformats embedded in web pages.

To wrap-up, the term "lowercase semantic web" have been coined by microformats proponents to counter the legacy "Semantic Web" (with capital "S" and "W"). The lowercase semantic web is not being proposed as an alternative to Semantic Web, the targets are quite different. While the latter aims at bringing full fledged reasoning capabilities to intelligent agents which will work for the human, lowercase semantic web aims to be an intermediate step, able to encode entities coming from everyday ontologies so that they can be accessed by everyday software.

In the spirit of lowercase semantic web, the W3C itself is proposing RDFa [Adida & Birbeck (2008)], an extension of XHTML which allows users to express semantics in Web pages. RDFa provides a set of attributes (hence the "a" in its name) to write statements directly mappable to RDF triples. Some of these attributes have already existed in XHTML (with the standards making them usable on all elements) and new ones have been introduced to specifically model RDF concepts: `about` (the resource a metadata refers to), `rel` and `rev` (forward/backward relationships), `typeof` (subject type), and few others. The fact that RDFa were designed over a well-defined model such as XML/RDF has important consequences on the language itself. RDFa supports namespaces opening interesting perspectives for embedding general and inter-mixed semantic information. Moreover it can exploits the data formats and data models proposed for RDF, which are very powerful to encode machine-readable knowledge. The possible presence of RDFa in future specifications of XHTML, as well as the support of standard bodies, can leverage the success of that solution. On the other hand, microformats already proved to be backed by several implementations and easily integrable in current systems, supported by a skilled and enthusiastic

5 http://microformats.org/wiki

6 http://creativecommons.org/license/

7 http://developer.mozilla.org/en/docs/Using_microformats

community of developers/users. Future developments, synergies or conflicts between these two proposals are then something to be monitored in the near future.

In conclusion, we observe that the lowercase semantic web seems to follow a 20-80 rule, with just 20% of the potential expressive power of Semantic Web it aims at implementing 80% of user semantic needs. For the time being it is going to be a success: just trying out tools like Operator, people get surprised to discover that Web pages authored by them support microformats out of the box, by the means of the tools they used to create them (e.g. the CC license chooser, and your favorite blog engine).

Semantic wikis

Now in parallel with the lowercase semantic web, the older effort of *semantic wikis* is trying to bridge the gap between easy authoring and semantic content for a different class of authors: wiki contributors.

Basically, a semantic wiki is a wiki system enabling users to write semantic data about a given domain. The idea is to join together the benefits of the wiki open editing model, and those of semantically-enriched content repositories. In fact, a semantic wiki is not (or, at least, it should not be) a complex authoring environment for ontologies, statements, semantic properties, etc. Rather, it is a wiki where authors can also write their content to be machine-readable.

Most semantic wikis adopt a simple yet powerful model: each page represents a concept of the domain, links can be typed to express relations among concepts, and properties of each concept can be defined via attributes. Specialized syntaxes are used to let authors write the semantic information. The point is to mitigate difficulties in writing semantic content by exploiting an authoring solution which proved to be powerful, flexible, and widely accepted by the Web community. Semantic wikis have given birth to a a lively and ever-growing community[8].

The example of Semantic Wikipedia [Völkel et al. (2006)] is significant. Its goal is to create a machine-readable version of Wikipedia, to better exploit the huge amount of available information. From a technical point of view, the project relies on Semantic MediaWiki, an extension of MediaWiki with a new syntax to let users annotate content fragments, and an enhanced interface to manage semantic data. Exportation capabilities to RDF/ OWL make the whole knowledge base available for advanced retrieving, searching, and reasoning. The technical robustness, the lively developer community, and the world-wide support and enthusiasm for Wikipedia make this semantic wiki project the most promising and solid.

Let us use a Semantic MediaWiki example to introduce the basic principle of most semantic wikis. The following snippet is the source code of a page about the seminal paper on Rhizome [Souzis (2005)].

```
The paper was presented at the [[accepted by::SemWiki2006]] as a
[[presented as::demo]]. It is about Rhizome.

* [http://www.liminalzone.org/static/semwiki2006-28.pdf Download]

Written by [[author::Adam Souzis]].
```

Some text fragments express semantic statements about that paper: for instance, the fact that the paper was written by `Adam Souzis` and accepted at `SemWiki2006` as a `demo`. Such information is not only readable by human users but also by sotware agents. Figure 1 shows the page as normally rendered by the wiki engine, and a table of automatically extracted semantic data.

The use of a new syntax is crucial to embed semantics. For instance, the fact that the paper is about Rhizome cannot be automatically retrieved since it was not correctly marked-up.

8http://www.semwiki.org/

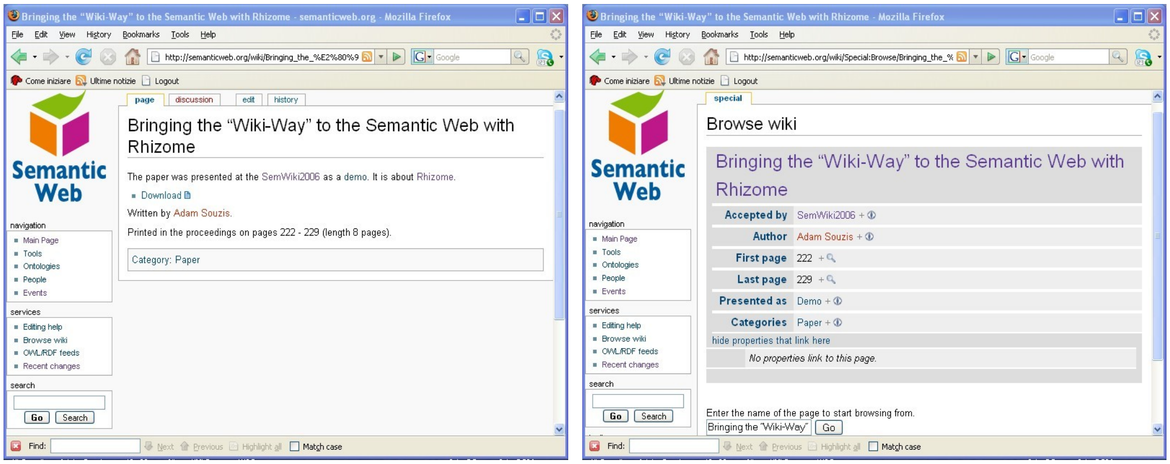

Figure 1. A Semantic WikiMedia page rendered by the engine and a connected page with some automatically extracted properties.

Note that the way to use a wiki does not change. In fact, users can freely edit any fragment without any limitation. The more the information is correctly encoded the more semantic data are available, but no constraint is imposed over the open wiki editing process.

Other semantic wikis are worth being discussed here. More than listing the peculiarities of each system, the goal is to highlight the most relevant aspects of a semantic wiki, and provide readers with metrics to evaluate existing and perspective solutions.

Two orthogonal dimensions can be used to classify a semantic wiki: *assisted editing* and *expressiveness*. Figure 2 organizes the most, in author opinion, relevant semantic wikis according to that classification. Each wiki is actually the representative of a wider class of wiki-clones all characterized by a specific approach in semantic content authoring.

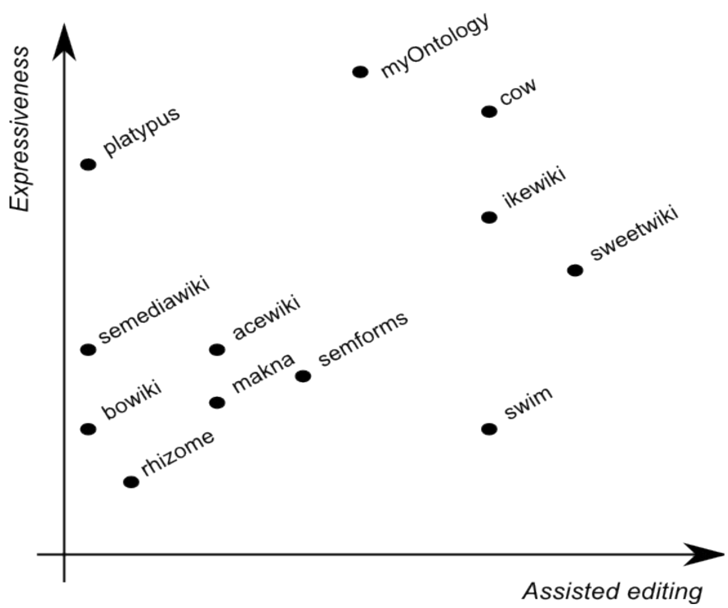

Figure 2. Showing semantic wiki clones on the axes of how much they assist/constraint authors (assisted editing), and how expressive they enable semantic annotations to be (expressiveness).

The *assisted editing* axis indicates how much a user is free to add or edit semantic data. Traditional wikis are based on a fully open editing model Leuf & Cunningham (2001) and owe their success to that idea. Most semantic wikis provide users alternative syntaxes for embedding semantic data within plain content using typed links. In that case, users do not have any limitations when writing semantic content: they keep on working on a plain textareas and keep on using the editing paradigm they are used to. Yet, they are not directly assisted in writing consistent declarations and statements, but the editing freedom of traditional wikis is saved. Platypus [Roberto Tazzoli & Campanini (2004)] (the first semantic wiki), Semantic MediaWiki (the most important competitor) and BOWiki [Backhaus et al. (2007)] (a domain-oriented extension of Semantic MediaWiki customized for the biology-domain) are examples of this approach.

While these wikis adopt a wiki-like syntax for inserting semantic data, others adopt XML-based syntaxes or their own languages. In a way, they limit the editing freedom of the users and drive them in writing content according to a given schema. That is the reason why Rhizome [Souzis (2005)] is slightly moved to the right in our plot. In fact, Rhizome relies on ZML (a textual syntax serializable into XML), a generic language to express semi-structured data, and an engine to apply rules for inter-mixing semantics and free texts.

In the middle of the *assisted editing* axis we found a class of wikis which integrates the traditional editing textarea with interfaces which help users to produce semantic data. AceWiki [Kuhn (2008)] exploits the Attempto controlled natural language ACE [Fuchs et al. (1998)] to let users write unambiguous statements in English. Although users can write inconsistent statements, the system also integrates a predictive authoring tool which suggests options and values to the users. Similarly, Makna [Dello et al. (2006)] presents a mixed interface where users can write content both with a wiki-like syntax and via specialized forms. Figure 3 shows two sample screenshots of this class of wikis, integrating text editors and semantic widgets.

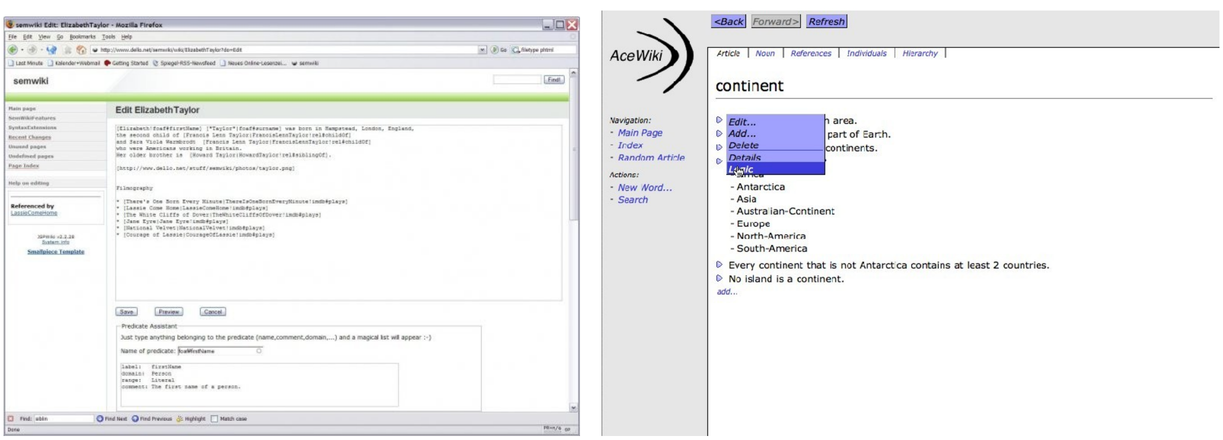

Figure 3. Makna, AceWiki and similar wikis integrate widgets for authoring semantic data with a plain text editor.

Semantic_Forms [Koren (2008)] also belongs to this group. It is an extension of Semantic MediaWiki, whose pages can be edited either via free textareas or via pre-defined forms. The system exploits templates, i.e. pre-defined text structures which are dynamically filled by content and rendered into a MediaWiki article. A form, in fact, is generated from a template, whose fragments and data unit have been previously typed. Since each data type is associated to a type of field (free textarea, checkbox or radio button with predefined values, predefined menu, etc.), a direct conversion process automatically produces the final form, to be filled by the users. The form-based content is indeed included into an article but the integration with the in-line content is only partial.

At the right edge of the spectrum we found those wikis which provide users powerful and highly-structured interfaces to add semantic data. IkeWiki [Schaffert (2006)] for instance has been designed to support expert users in authoring ontologies. These tools are in fact defined "ontology editors". It is significant the fact that IkeWiki provides distinct interfaces to edit plain wiki content, metadata, and annotations. The COW's interface [Fischer et al. (2006)] is clearly oriented to traditional concepts of ontology management such as classes, relations, instances, …. SweetWiki [Buffa & Gandon (2006)] is at the very right of the spectrum since it integrates Ajax-based widgets to add metadata and edit content. Although these interfaces are very intuitive, they limit the editing freedom of the users which are driven *step-by-step* in writing semantic content. Figure 4 shows two SweetWiki's screenshots, as examples of these ontology-oriented interfaces.

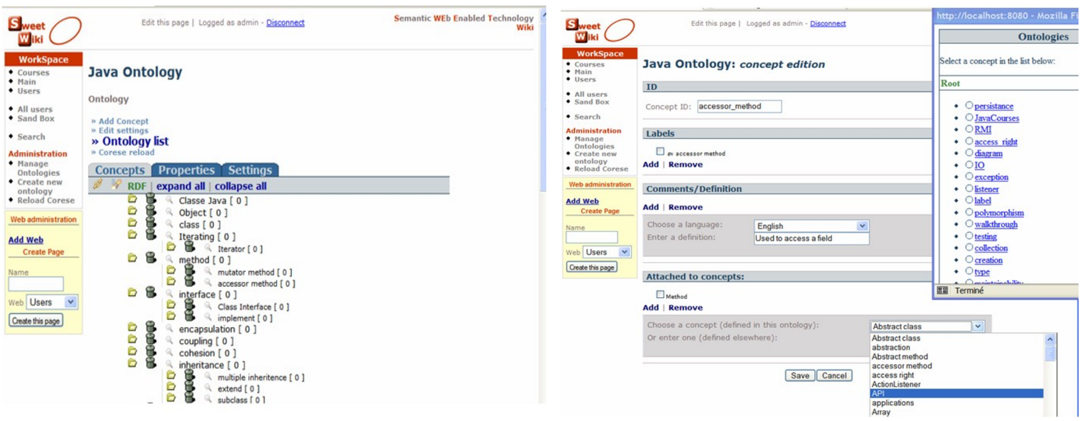

Figure 4. The SweetWiki interface, like other 'ontology editors' includes modules for advanced management of ontologies.

The orthogonal dimension—*expressiveness*—indicates how much a wiki user can alter the ontology encoded in the wiki. The idea is to evaluate whether the users are actually able to describe their domain: can they add new classes and instances? which types of relations can be declared? which types of constraints? which statements can be directly written or derived?

Although being a very flexible framework, Rhizome [Souzis (2005)] sits at the bottom of our plot. It represents all those wikis which allow users to express RDF statements but do not let them to build deep ontologies with new classes, new sub-classes relations, …. BOWiki [Backhaus et al. (2007)] and SWIM [Lange (2007)] (an extension of IkeWiki customized for collaborative management of scientific knowledge) are examples of wikis relying on built-in ontologies. These wikis allow users to add new instances and provide them with partial support for creating new classes, but they are still tied to a specific domain. That is the reason why they are in a lower position, with respect to Semantic MediaWiki, AceWiki, and Semantic_Forms. All these wikis, in fact, allow users to create any class (category) useful in their domain, and allow any type in the definition of links. On the other hand, they adopt a very simple ontological model which maps each concept to a page: more complex and deep ontologies cannot be created, as well as more complex statements and inferences are not available. The subject of each statement, for instance, can only be the concept or instance represented by a page, no finer-grained mechanism is provided.

It is not a surprise the fact that the ontology editors (IkeWiki, COW, and to a lesser extent SweetWiki) are positioned near the top of the plot. They in fact provide users interfaces to build complex ontologies, to express complex relations and to interact with OWL reasoners. On the contrary, the position of Platypus was unexpected. The reason is that Platypus, like other semantic wikis, allows users to write complex RDF statements and goes beyond the basic ontological model of Semantic MediaWiki.

The very top position on this axis is occupied by MyOntology [Siorpaes & Hepp (2007)]. That is not only a semantic wiki-clone, rather a wider project aiming at defining theoretical foundations for the design and implementation of community-driven and wiki-based ontology editors. MyOntology represents all those approaches using wikis as platforms for creating and managing (even very complex) ontologies during their whole lifecycle, being able to model any type of class, instance, relation and constraint.

The two dimensions appear to be correlated. The general trend is that structured and ontology-oriented interfaces give users more power but limit their editing freedom. On the contrary, simplified editing models tend to limit their "semantic capabilities". Exceptions and trade-off solutions exist and have been noticed. However, the tension between a free editing model and formal/rigid semantic authoring is still evident within semantic wikis and none of the existing solutions maximizes both perspectives.

WIKI TEMPLATING AND THE SEMANTIC WEB

The above-discussed efforts to make it easier to author metadata within web pages are very promising. Each of them, however, is targeted to a given context and in most cases to a given class of users. An open question is then: is there an intrinsic and universal trade-off between semantically rich content and ease of authoring? What is the best solution to solve it?

We believe that wikis are the best candidates to play that role, for two reasons: first of all, wikis have proved to be simple enough for inexperienced users to create large content repositories such as Wikipedia and are already well-known and established; second, wikis are designed for (even huge) communities and can exploit the expertise and enthusiasm of such communities to provide semantic information. The role of the community is essential for our purposes. While blogs are meant for being edited by a single user and read by many others, in fact, wikis allow all users to share their thoughts and ideas and make it possible to build richer and larger knowledge-bases. This does not mean that it is pointless to develop techniques and technologies for ease authoring of semantic content in the context of simplified CMS, but only that on that area the benefits of the community can not be similarly exploited. For these reasons, in the reminder of this chapter we will focus on wikis, inviting the reader to develop parallel considerations for the microformat/blog side.

Requirements

The question we have posed can then be rephrased as follows: *what are the requirements for a wiki which enables semantic data authoring?* We propose 4 such requirements:

editing freedom users should be able to freely edit any content in an environment as unconstrained as possible. The key point is to let authors to freely edit not only the textual content as it happens in legacy wikis, but also the semantic content.
Such an open paradigm determined the success of wikis: it is natural to expect it for semantic wikis too.

content metadata proximity semantic information should be as close as possible to content.
The common approach to add metadata (that is: data about data) to human-readable documents on the Web is to create a machine-readable serialization of metadata, to be published either as an external resource or as a special-purpose section of the original document. From the point of view of the author, this approach can induce an information duplication: the same information got inlined in prose for humans *and* in more rigorous languages for machines. Two problems arise, the first is inherent to information duplication when not paired with coherence enforcement: the semantic content can easily get out-of-sync with respect to the prose. The second problem is that the authoring process becomes more tedious, since diligent content authors striving for coherence need to edit two documents instead of one. This is made even more tedious by the fact that the documents need to be written in different languages, possibly requiring entirely different

technologies or knowledge.

validation it should be possible to check for correctness the semantic information with respect to user needs and preferences, as well as domain requirements.
This requirement is not brought by semantic annotations. The presence of semantic information, however, emphasizes it for two reasons: (i) semantic agents can now perform advanced controls over content; (ii) valid content makes possible advanced searching and reasoning processes. Note that correctness contributes to avoid wasting author efforts in adding semantic annotations: malformed or incoherent semantic annotations are as useless as no semantic annotation at all. Note also that the support for validation should be independent from a set of specific requirements: regardless of the actual checks users are interested in, a semantic wiki should help them to declare those requirements and to verify the quality of their metadata.

uniformity the fact that we are dealing with semantics in an open editing environment suggests an uniformity requirement. Semantic information allows users to make sense of an object in relation to others of the same kind, other versions or variants of it, other external resources, etc. The overall objective is building an uniform knowledge base, where similar objects are easily identifiable, (common) properties are retrievable, and objects are related.
Generating such an uniform knowledge-base via exportation from authoring tools which constraint the authoring work-flow (e.g. using a CMS) is not particularly challenging. Such tools can indeed force authors to provide “required” information and keep them under tight control. How useful is the resulting knowledge-base is an entirely different topic, but at least uniformity can be inherited from authoring-time constraint. Achieving similar results with wikis is more difficult, due to their open editing model and the heterogeneity of involved users. Wiki users would benefit from a mechanism aiding them in producing uniform pages sharing content and structures.

This section looks at semantics wikis and review them with respect to the above requirements: do semantic wikis meet the requirements? We believe they still fail in satisfying *all* of user needs, as each of them is particularly strong on some aspects but has important weaknesses on others.

The analysis about the *assisted editing* property in the previous section highlighted that the *editing freedom* requirement is neglected by several state of the art wikis: it is very common to find semantic wikis implementing free editing interfaces for non-semantic contents, but restricting (sometimes heavily) author freedom in authoring semantic annotations. They were all shown on the right of Figure 2. Other semantic wikis fail in reducing the *content/metadata distance*, since they only partially embed semantic data within the content. Semantic_Forms for instance provide form-based interfaces to insert semantic data, but consider those data as independent fragments. The most critical point is however *validation*. Semantic wikis in fact have a very poor support for automatic data-checking. Semantic MediaWiki and Semantic_Forms for instance verify whether a mandatory property is included in a template-based page, or whether it belongs to a given category. On the other hand, they do not support controls on semantic data which crosses page boundaries, like relations expressed as typed-links between pages. Similarly, other wikis validate data inserted using specialized interfaces but do not support validation on in-line content. The main issue, in fact, is that validation is still limited to highly-structured data which are handled as an extra layer disconnected from the free text users can (and want to) edit.

This chapter proposes an alternative solution based on *wiki content templates*. The idea of wiki templates is not new in the literature, but one must be careful: here the focus is specifically on *content* templating, i.e. mechanisms that enable users to define content snippets that can be invoked from other pages, possibly instantiating some of their parts.[9] Content templates are meant to reuse wiki content across different pages, and to ease the creation of similar pages. Examples of such templates are very common:

9the alternative interpretation of templating in the wiki context is *presentational templating*, such as HTML frames or CSS stylesheets used for page rendering

Wikipedia supports templates for creating uniform page parts (e.g. “infoboxes”) which provide the same information set on nations, actors, bands, players, …; seeding pages are supported by many wiki clones to create new pages about recurring events, information, … The strength of templates lies in their capacity of modeling information patterns and boiling down multiple editing actions into a single one. They help users in saving time and easily produce high-quality results, exploiting polished structures. This chapter proposal is to foster the creation of semantic information through semantically-enhanced templates, as like as templates proved capable to foster the creation of plain information via plain templates. The key point is that templates allow users to do so without changing the wiki editing work-flow authors are already familiar with.

Templating flavors

An analysis [Di Iorio et al. (2008)] of exiting wiki templating models pointed out some aspects of templating, which ends up being central in the design of the framework implementing this chapter proposal. The authors identified two approaches at content templating: functional and creational. A *functional template* is a page including a set of placeholders which will be substituted by actual values passed as formal parameters at template application time. Such a template is applied invoking it by name and passing actual parameters, with a special-purpose syntax in the wiki markup. For instance, most Wikipedia information boxes providing structured information about sports, animals, and plants are created using such technique. The following code snippet shows the invocation of a functional template in a page about ’Italy’.

```
{{Infobox Country
|native_name        = ''Repubblica Italiana''
|capital            = [[Rome]]
|government_type    = [[Parliamentary republic]]
|leader_title1      = [[President of the Italian Republic|President]]
|leader_name1       = [[Giorgio Napolitano]]
...
}}
```

The author has to only specify the template name (`Infobox Country`) and the properties of this country (the capital is Rome, the ’President’ is currently ’Giorgio Napolitano’, etc.). Whenever that page will be accessed, such declaration will be rendered as a table with a proper style, summarizing all these properties.

A *creational template* (or seeding page) is a page used as the starting content for the creation of new ones, with the same initial structure and markup, as if it had been copy&paste-d into the new page. Usually, when creating new pages, users of creational template enabled wiki engine, are faced with the list of available templates they can choose from. Though seeding pages were available in the early wiki days, MoinMoin first re-introduced in modern engines such templating model, which is now re-gaining momentum. Creational templates are widely used to create (initially) uniform pages in a wiki site or to quickly generate new content from pre-existing one. A creational template for the ’Italy’ example would be similar to the following (syntactical details, i.e. square-brackets, are not relevant here):

```
[[Insert the name of your country]] was natively
called [[insert the native name of your country]].
The capital is [[Insert the capitol here]].
It is a [[Insert the type of the government]],
whose leader is titled [[Insert the title of the
country's main leader]]. The current leader is
[[Insert the name of the current leader]]....
```

Our proposal is to include partial semantic information in a template, which can be instantiated and completed by the users. Several reasons make creational templates the most appropriate choice to satisfy the above requirements. First, they better meet the requirement of reducing the distance between content and metadata (requirement 2). The semantic information provided in a creational template is *fully embedded* with the textual content of that page. There is in fact a one-to-one mapping between the structure of the template and the structure of the rendered page, so that the information is also easier to be searched, retrieved and updated.

On the other hand, the rendered text derived from a functional template can have a completely different structure from the text edited by the user (usually that is a simple invocation of a function by passing parameters). Such a different organization have in fact some important benefits: functional templates make it easy to present structured data and to force a set of pages to be uniform. Moreover, the rendering of *all* the pages connected to the same template can be updated with a single operation. In many cases functional templates are useful and appropriate: the experience of Semantic Wikipedia and especially Semantic_Forms proved they also are a precious aid for semantic wiki authoring. Nevertheless functional templates fail to meet the editing freedom requirement. Although filling an instance of a functional template is very simple, users are not free to edit their (semantic) content. They can only add data to a pre-defined structure which will be rendered later, instead of having the possibility of re-organize and customize their content or information pattern. A partial solution has been proposed (by MediaWiki, for instance) that allows users to mix free-text with fragments rendered through a functional template[10] . However, those fragments are not completely integrated with the mixed textual content: they duplicate the information, and are located in a separated and independent area of the page. This issue is then related to the requirement of fully embedding semantic content into a page, in order to provide an integrated yet open editing environment. Creational templates provide such an environment.

Still, creational templates have an important limitation: once a page has been derived from a template, it lives as an independent entity within the wiki. It can be modified up to become something completely different from the original source. The idea of using a template to drive the authoring of semantic information in a page, and validate it (requirement 3, validation) is then not viable by only adopting creational templates. Our solution to this remaining problem is then to extend creational templates to also support validation and guarantee uniformity (requirement 4, uniformity).

Light constraints

The extension of creational templates with validation can be achieved exploiting the general-purpose architecture of *lightly constrained wikis* [Di Iorio & Zacchiroli (2006)] (LCWs). The idea of LCWs is rooted at the observation that some constraints spontaneously appear in wiki communities to encode best authoring practices or enforce domain-specific features: capabilities of spell-checking, detecting orphan-pages, giving the same structure to a set of pages, and forcing the presence of some information are all instances of this habit. LCW is a general framework, which integrates a non-invasive mechanism to validate a posteriori page content, so as to automatically check whether the constraints on a given page are violated or not, notifying users of problems without forcing authors to fix them.

The architecture of a LCW is simple: each page is associated to a set of validators, each validator implements a constraint-check and returns a validation report, possibly completed by a list of localized errors. In a LCW, traditional Save and View operations turn into: ConditionalSave (whenever a user tries to save a page, validation reports are presented and the author can ignore them saving the page anyway or rather fix them by editing the page again) and AnnotatedView (visitors viewing a page are shown the usual content enriched by a validation report, so that they can help in fixing errors). The most important aspect of LCWs is their *lightness*, the fact that constraints do not have to be necessarily satisfied in order

10 http://www.mediawiki.org/wiki/Help:Templates

to save a page, but are conceived as warnings. Such a solution does not impose any limitation over the editing process, preserves “The Wiki Way” Leuf & Cunningham (2001), and relies on the community (the true wiki power) to fix errors by only making them explicit.

Semantic wiki templating

Going back to the initial issue, we propose to integrate easy authoring and semantic data by exploiting creational templates empowered by light constraints on template matching: we call this solution *lightly constrained templates* (LCT). The basic idea is to (i) write templates whose fragments contain partial semantic information to be filled by users and (ii) encode as a light constraint the fact that a page matches that template. Whenever a page is displayed or saved, the system verifies whether the page still matches the template (or, at the very minimum, whether its semantic fragments are still preserved in that page) and presents a validation feedback according to the above-discussed work-flow.

The synergy between creational templates and light-constraints validation provides several benefits. First of all, users do not suffer any limitation in editing (semantic) content, since constraints can be temporarily violated and the whole content is available for modification. Unlike wikis which provide interfaces driving users and restricting their editing choices *a priori*, LCT validate content *a posteriori* without any further imposition on authoring. Requirement 1 (editing freedom) is hence dealt with. The fact that templates are creational, and so there is a one-to-one correspondence between the structure of the source and the structure of its rendering, satisfies requirement 2 (content-metadata proximity). LCT provides a unified editing environment, unlike most semantic wikis which offer disconnected interfaces to work on prose or semantic data. The difference with other wikis which offer in-line editing of semantic data lies in the validation capabilities: the possibility to add multiple validators, each providing localized analysis of content make possible to implement a plethora of controls. The more validators are powerful, the more (semantic) data can be checked and managed. In fact, requirement 3 (validation) and 4 (uniformity) are satisfied by the presence of external validators which check whether or not the originating template is matched and the (semantic) information is preserved and uniformed.

A piece is still missing in the discussion, about mechanisms used by the authors to write semantically-enabled creational templates. The analysis of the background section suggests two solutions. On one hand, an enhanced wiki markup can be exploited (Semantic Mediawiki, for instance, uses an extension of the plain Mediawiki syntax to express typed links, categories, and relations among pages). On the other hand, microformats provide solutions for including semantic information within HTML pages: the fact that most wikis are able to parse HTML source code then suggests to use *quasi* microformat syntax in creational templates. Still, actual syntaxes are not that relevant at this stage: what is important is that semantic content is embedded and can be validated a posteriori.

The nature itself of the validation is something which should not be fixed once and for all. In fact, LCT is designed to support different *levels of validation*. Multiple types of matching between an instance and its originating template exist and can even coexist (since multiple validators can be associated to a single page). Consider a wiki page about a music band containing free text and embedded semantic information about its components and albums in two distinct lists. That page can be derived from a creational template which generates a first instance with some holes for the semantic data (and suggestions to add more data of the same type). New items can be added to the lists of albums and components and the free text can change during the life cycle of that page. A validator which verifies that all the originating text is preserved would return a warning as soon as a single character changes in the free text area; on the other hand, a validator only checking if semantic fragments in the lists match the originating template would ignore free text modifications. The point is that some changes impact on (part of) the semantic content of a page, others do not: proper designed validators can then implement different strategies for template matching.

A template is actually a ’schema’ defining the semantic information to be provided in a page. Two pages with very different textual content (or organized in very different structures) could both match the same template if they provide the same set of semantic data, as required by the template. It is important to notice that these pages do not have to contain exactly the same data but they have to be two instances of the same class.

Note also that users are still entitled to change *any* fragment of that page, without *any* limitation: validators will return an error/warning only if a required semantic information is missing, corrupted or incongruous.

FUTURE RESEARCH DIRECTIONS

The research in the field of simplified semantic wiki authoring is far from being complete. The presented proposal plans some important developments. The first available prototype [Di Iorio et al. (2008)] is an extension of MoinMoin (see Figure 5 for a screenshot), a wiki clone providing creational templates, which implements the LC architecture and provides a simple support for template validation. A template organizes content into hierarchical structures, so that the instance/template matching consists of checking that all (sub-)sections exist and provide some basic information. Such a preliminary mechanism need to be extended to validate in-line content and relations among resources.

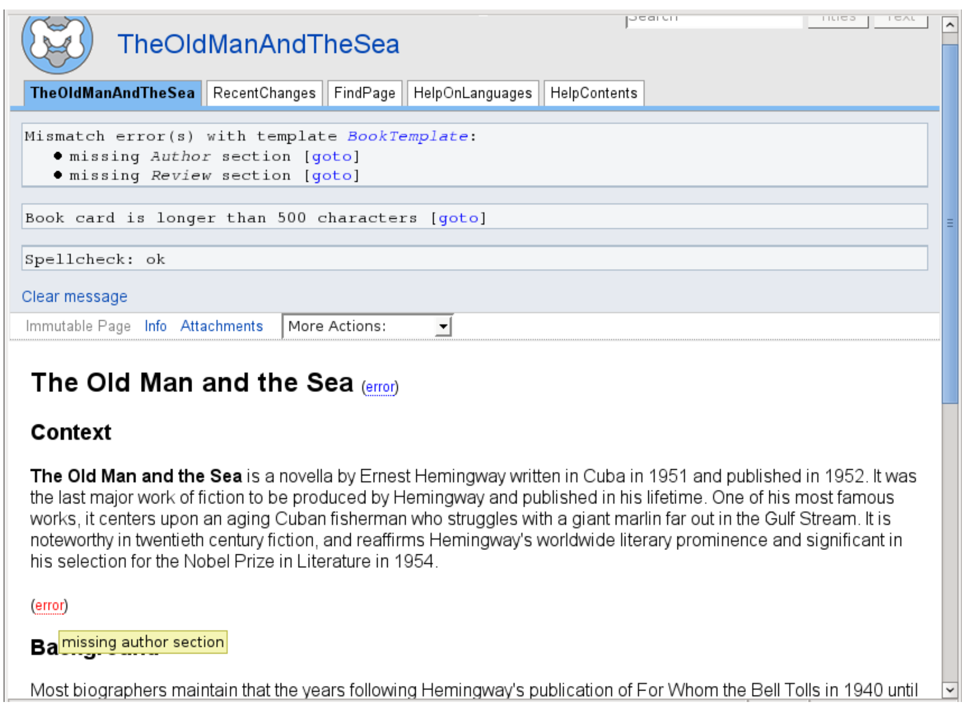

Figure 5. Screenshot of an LC templating implementation for the MoinMoin wiki engine. The shown tooltip is associated with the first of the template mismatch errors summarized at the top of the page, localized at the second `(error)` *marker in the page*

.

A key aspect of the proposed architecture is the independence from the wiki platform it applies to. Decoupling validators from the wiki engine makes it possible to re-use validators among different clones, to implement multiple validation strategies and to code validators separately. One of the needed next steps is coding extensions for other wikis (for instance, MediaWiki which currently supports functional templates and only partially creational templates) and adapting the general-architecture to different domains and use-cases.

Moving away form implementation issues, a research issue is central to the proposed vision: embedding semantic data into a wiki template. Though from different perspectives, we believe it will be further studied by the community. In the field of semantic wikis, in fact, studies on simplified syntaxes for semantics are still alive and researchers will probably end up discussing a unified solution, as well as they have done for the plain one [Sauer et al. (2007)]. The same liveness is evident in the microformat community, where new formats are continuously introduced and supported by server- and client-side applications. These trends need to be monitored and need contribution, in order to find a flexible and powerful syntax to input semantic annotation inline in wiki markup. The increasing importance of WYSIWYG editors, on the other hand, can help in hiding complex syntactical aspects. The complex relation between assisted interfaces and traditional wiki textareas is another aspect to be investigated from the point of view of human-computer interaction.

The expressiveness of the semantic language also deserves attention. It does not only impact on the syntax, but also on the overall capabilities of the system. In fact the overall integration between simple authoring systems and semantics requires radical simplifications. It is also true, however, that richer and more powerful semantic information could be very useful. Most semantic wikis, for instance, model each entity of a domain as a wiki page and force users to only insert in a page statements about that page. Do different scenarios exist where users benefit from a model where page fragments express statements about external resources? If yes, is it possible to support that model without complicating too much the editing work-flow? What about efficiency issues in retrieving such distributed information? These are only a few questions about the very same issue: the expressive power we need to give to our users.

The power of the language impacts on the complexity of the validation process too. Validation issues can play a leading role in future research pivoting around semantic content management systems. The modularity of the proposed architecture allows to incrementally add validators and experiment with novel checks, foreseeable topics are: validation of in-line content, cross-relations, and accurate analysis of free text. The coding and programmability of each validator is also a central issue. A validator is an independent component which can be invoked either as an internal function by the wiki, or as an external service. Work is needed both on tight integration possibilities with wiki clone (up to the possibility of coding validators directly within wikis) and on loose integration possibilities to formalize interfaces and interaction within wikis and external validators. The proposed framework distinguishes between two roles for the users, in order to only require few of them to manage templates and validators: template authors (or *tailors*) and ordinary users; further studies on the role of the tailors, as well as mechanisms they can use to drive semantic information insertion, is yet another "plug" to the sketched research.

Finally, the blog/microformats side can (and should) be studied to highlight analogies and differences with the presented analysis. To start with, the review of how microformats score against the requirements of background section is open (as well as the question whether the proposed requirements need better tuning due to the change of environment …). The initial considerations are that metadata proximity is properly implemented by microformats, editing freedom is likely not that relevant (as blogs are usually targeted at a single person, which can enjoy a relative high degree of freedom in her doings). Can one hope to achieve uniformity in the blogosphere made by independent authors working on Web applications completely decoupled one from the other? In this respect a single wiki system looks like a more closed universe easier to tame. What about validation? Once more one cannot hope to deploy a validation framework only once, but should rather rely on well-known access points as blog feeds to perform remote analysis as independent entities like Technorati are already doing. All these considerations are very preliminary and deserves further investigation, each of them has the potential to open relevant topics for related research.

CONCLUSION

The two most recent evolutions of the World Wide Web—Semantic Web and Web 2.0 —are characterized by a very interesting dichotomy: what is a strength point for one seems to be a difficult-to-be-crossed barrier for the other, and viceversa. The Semantic Web is meant to be a machine readable platform for advanced information search and retrieval, but has had difficulties in taking off since that information is still difficult to be authored; on the other hand, the Web 2.0 stresses easy authoring for all users, but does not rely on consistent and complete semantic information.

This chapter has investigated solutions to bridge the gap between these two worlds: both the lowercase semantic web and semantic wikis has been discussed in that respect. The former is a concept indicating all those proposals which are not as complex or extensible as the languages related to the traditional Semantic Web (such as RDF and OWL), but let authors add simple semantic content and processable annotations to Web pages. In particular, the chapter presented the key characteristics of microformats which exploit legacy XHTML attributes to convey semantics onto Web pages. On the other hand, semantic wikis combine the power of free wiki editing with semantic knowledge-bases, by giving to users alternative syntaxes to annotate content, as well as integrated interfaces to search and browse that content.

A detailed analysis of the limitations of semantic wikis has led to a novel alternative solution based on wiki content templating. The basic idea is to piggyback semantic information onto content templates, and extend semantic wikis to support (semantic) content validation. Details of that solution were discussed in the core part of the chapter. The resulting discussion identified several research directions, related not only to the proposed solution but also to the whole area of simplified semantic web authoring.

Starting from here, the interested reader is invited to investigate parallels and differences between the proposed wiki-based scenario and the microformats scenario, intimately related to their usage in blogs. The adoption of XHTML extensions to write wiki content, as well as the integration of semantic data from different sources (wikis and blogs), or the application of templates to the blog scenarios are only some of the possible starting points. The investigation and merge of similar experiences can lead to a unified authoring environment which will turn the dream of the Semantic Web into an everyday reality.

REFERENCES

Adida, B. & Birbeck, M. (2008). RDFa primer: embedding structured data in web pages. W3C Working Draft 17 March 2008, http://www.w3.org/TR/xhtml-rdfa-primer/.

Alden, R. H. & Bartlett, S. J. (1996). vCalendar: The electronic calendaring and scheduling exchange format. www.imc.org/pdivcal-10.doc. versit Consortium, v.1.0.

Backhaus, M., Kelso, J., Bacher, J., Herre, H., Hoehndorf, R., Loebe, F., & Visagie, J. (2007). Bowiki – a collaborative annotation and ontology curation framework. In *Proceedings of Workshop on Social and Collaborative Construction of Structured Knowledge*.

Barlow, A. (2007). *The Rise of the Blogosphere*. Praeger Publishers.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, *284*, 28–37.

Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V., & Oinas-Kukkonen, H. (1997). Fourth generation hypermedia: some missing links for the world wide web. *Int. J. Hum.-Comput. Stud.*, *47*(1), 31–65.

Buffa, M. & Gandon, F. (2006). Sweetwiki: semantic web enabled technologies in wiki. In *WikiSym ’06: Proceedings of the 2006 international symposium on Wikis*, (pp. 69–78)., New York, NY, USA. ACM Press.

Connolly, D. & Miller, L. (2005). RDF calendar - an application of the resource description framework to iCalendar data. http://www.w3.org/TR/rdfcal/. W3C Interest Group Note.

Dello, K., Simperl, E. P. B., & Tolksdorf, R. (2006). Creating and using semantic web information with makna. In Völkel, M. & Schaffert, S. (Eds.), *Proceedings of the First Workshop on Semantic Wikis – From Wiki To Semantics*, Workshop on Semantic Wikis. ESWC2006.

Di Iorio, A., Vitali, F., & Zacchiroli, S. (2008). Wiki content templating. In *WWW2008: Proceedings of the 17th International World Wide Web Conference*. To appear.

Di Iorio, A. & Zacchiroli, S. (2006). Constrained wiki: an oxymoron? In *WikiSym '06: Proceedings of the 2006 international symposium on Wikis*, (pp. 89–98)., New York, NY, USA. ACM Press.

Fischer, J., Gantner, Z., Rendle, S., Stritt, M., & Thieme, L. S. (2006). Ideas and improvements for semantic wikis. *Lecture Notes in Computer Science*, *4011 / 2006*.

Fuchs, N., Schwertel, U., & Schwitter, R. (1998). Attempto controlled english - not just another logic specification language. In *LOPSTR*, (pp. 1–20).

Hendler, J. (2008). Web 3.0: Chicken farms on the semantic web. *Computer*, *41*(1), 106–108.

Khare, R. (2006). Microformats: The next (small) thing on the semantic web? *IEEE Internet Computing*, *10*(1), 68–75.

Koren, Y. (2008). Semantic forms. http://www.mediawiki.org/wiki/Extension:Semantic_Forms.

Kuhn, T. (2008). Acewiki: A natural and expressive semantic wiki. In *Proceedings of Semantic Web User Interaction at CHI 2008 (to appear)*.

Lange, C. (2007). Towards scientific collaboration in a semantic wiki. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, (pp. 119–126).

Leuf, B. & Cunningham, W. (2001). *The Wiki Way: Quick Collaboration on the Web* (Pap/Cdr ed.). Addison-Wesley Professional.

Manola, F. & Miller, E. (2004). RDF primer. http://www.w3.org/TR/rdf-primer/. W3C Recommendation.

McGuinness, D. & van Harmelen, F. (2004). OWL web ontology language overview. http://www.w3.org/TR/owl-features/. W3C Recommendation.

Musser, J. & O'Reilly, T. (2006). *Web 2.0 Principles and Best Practices*. O'Reilly Media. O'Reilly radar report.

O'Reilly, T. (2007). What is web 2.0: Design patterns and business models for the next generation of software. *Communications and Strategies*, *65*(1), 17–38.

Prud'hommeaux, E. & Seaborne, A. (2008). SPARQL query language for RDF. http://www.w3.org/TR/rdf-sparql-query/. W3C Recommendation.

Roberto Tazzoli, P. C. & Campanini, S. E. (2004). Towards a semantic wiki wiki web. In *3rd International Semantic Web Conference (ISWC2004)*.

Sauer, C., Smith, C., & Benz, T. (2007). Wikicreole: a common wiki markup. In *WikiSym '07: Proceedings of the 2007 international symposium on Wikis*, (pp. 131–142)., New York, NY, USA. ACM.

Schaffert, S. (2006). Ikewiki: A semantic wiki for collaborative knowledge management. In *WETICE '06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, (pp. 388–396)., Washington, DC, USA. IEEE Computer Society.

Siorpaes, K. & Hepp, M. (2007). myontology: The marriage of ontology engineering and collective intelligence. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, (pp. 127–138).

Souzis, A. (2005). Building a semantic wiki. *IEEE Intelligent Systems*, *20*(5), 87–91.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., & Studer, R. (2006). Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006*.

ADDITIONAL READINGS

Adida, B. (2008). hgrddl: Bridging microformats and rdfa. *Web Semant.*, *6*(1), 54–60.

Allsopp, J. (2007). *Microformats: Empowering Your Markup for Web 2.0*. Friends of Ed.

Ankolekar, A., Krötzsch, M., Tran, T., & Vrandecic, D. (2007). The two cultures: mashing up web 2.0 and the semantic web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, (pp. 825–834)., New York, NY, USA. ACM.

Auer, S. & Lehmann, J. (2007). What have innsbruck and leipzig in common? extracting semantics from wiki content. *The Semantic Web: Research and Applications*, 503–517.

Bush, V. (1996). As we may think. *interactions*, *3*(2), 35–46.

Cayzer, S. (2006). What next for semantic blogging? In *Proceedings of Semantics 2006*.

CosmoMode. WikiMatrix: compare them all. http://www.wikimatrix.org/.

Flores, F. C., Quint, V., & Vatton, I. (2006). Templates, microformats and structured editing. In *DocEng '06: Proceedings of the 2006 ACM symposium on Document engineering*, (pp. 188–197)., New York, NY, USA. ACM.

Greaves, M. (2007). Semantic web 2.0. *IEEE Intelligent Systems*, *22*(2), 94–96.

Haake, A., Lukosch, S., & Schümmer, T. (2005). Wiki-templates: adding structure support to wikis on demand. In *WikiSym '05: Proceedings of the 2005 international symposium on Wikis*, (pp. 41–51)., New York, NY, USA. ACM Press.

Heath, T. & Motta, E. (2008). Ease of interaction plus ease of integration: Combining web2.0 and the semantic web in a reviewing site. *Web Semant.*, *6*(1), 76–83.

Karger, D. R. & Quan, D. (2005). What would it mean to blog on the semantic web? *Web Semantics: Science, Services and Agents on the World Wide Web*, *3*(2-3), 147–157.

Khare, R. & Çelik, T. (2006). Microformats: a pragmatic path to the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, (pp. 865–866)., New York, NY, USA. ACM.

Koren, Y. (2008). Semantic wiki state of the art. http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art.

Krug, S. (Ed.). (2000). *Don't Make Me Think! A Common Sense Approach to Web Usability*. New Ryders.

Lange, C., Schaffert, S., Skaf-Molli, H., & Völkel, M. (Eds.). (2008). *Proceedings of the 3rd Semantic Wiki Workshop (SemWiki 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, volume 360 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Lassila, O. & Hendler, J. (2007). Embracing "web 3.0". *IEEE Internet Computing*, *11*(3), 90–93.

Millard, D., Bailey, C., Boulain, P., Chennupati, S., Howard, Y., Davis, H., & Wills, G. (2008). Semantics on demand: Can a semantic wiki replace a knowledge base? *New Review of Hypermedia and Multimedia*, *14*(1).

Möller, K., Bojars, U., & Breslin, J. G. (2006). Using semantics to enhance the blogging experience. In Sure, Y. & Domingue, J. (Eds.), *ESWC*, volume 4011 of *Lecture Notes in Computer Science*, (pp. 679–696). Springer.

Passant, A. & Laublet, P. (2008). Towards an interlinked semantic wiki farm. In Lange et al. (2008).

Quint, V. & Vatton, I. (2007). Structured templates for authoring semantically rich documents. In *SADPI '07: Proceedings of the 2007 international workshop on Semantically aware document processing and indexing*, (pp. 41–48)., New York, NY, USA. ACM.

Rauschmayer, A. (2008). Next-generation wikis: What users expect; how rdf helps. In Lange et al. (2008).

Schmedding, F., Hanke, C., & Hornung, T. (2008). Rdf authoring in wikis. In Lange et al. (2008).

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, *21*(3), 96–101.

WikiIndex. WikiIndex: the wiki of wikis. http://www.wikiindex.org/Welcome.

KEY TERMS & DEFINITIONS

Microformat: a class of mark-up languages to embed semantic data into web pages, by exploiting XHTML attributes and elements.

Semantic Web: an extension of the World Wide Web, aiming at defining the web content as a machine-understandable information which can be searched, collected and managed by software agents.

(lowercase) semantic web: an intermediate step towards the (uppercase) Semantic Web, aiming at expressing semantic data within HTML pages in a simple and effective way.

Semantic wiki: a wiki enabling users to write and manage semantic information about a given domain.

Template: a description of a layout and the rules to produce that layout from the input content.

Web 2.0: a term used to indicate the recent (economic, technical and social) trends in the World Wide Web, stressing on information sharing, collaboration, personalization and social connectivity.

Wiki: a collaborative web editing environment for shared writing and browsing, allowing every reader to access and edit any page.

Wiki clone (or "Wiki engine"): a software, written in a specific programming language, that runs a wiki.