

# Annotation and Navigation in Semantic Wikis<sup>\*</sup>

Eyal Oren<sup>1</sup>, Renaud Delbru<sup>1</sup>, Knud Möller<sup>1</sup>, Max Völkel<sup>2</sup>, and Siegfried Handschuh<sup>1</sup>

<sup>1</sup> DERI Galway, Ireland

`firstname.lastname@deri.org`

<sup>2</sup> Forschungszentrum Informatik, Karlsruhe, Germany

`voelkel@fzi.de`

**Abstract.** Semantic Wikis allow users to semantically annotate their Wiki content. The particular annotations can differ in expressive power, simplicity, and meaning. We present an elaborate conceptual model for semantic annotations, introduce a unique and rich Wiki syntax for these annotations, and discuss how to best formally represent the augmented Wiki content. We improve existing navigation techniques to automatically construct faceted browsing for semistructured data. By utilising the Wiki annotations we provide greatly enhanced information retrieval. Further we report on our ongoing development of these techniques in our prototype SemperWiki.

## 1 Introduction

Wikis are collaborative hypertext authoring environments. Wikis allow people to collaboratively collect, describe, and author information. Since most information in ordinary Wikis consists of natural-language texts, structured access and information reuse are practically not possible [13].

Semantic Wikis allow users to make formal descriptions of resources by annotating the pages that represent those resources. Where a regular Wiki enables users to describe resources in natural language, a Semantic Wiki enables users to additionally describe resources in a formal language. By adding metadata to ordinary Wiki content, users get added benefits such as improved retrieval, information exchange, and knowledge reuse.

An ordinary Wiki should offer functionality<sup>3</sup> such as access control, binary data management, version management, notification, and data export. In our opinion, a Semantic Wiki should specifically address three additional questions:

1. how to annotate content?
2. how to formally represent content?
3. how to navigate content?

---

<sup>\*</sup> This material is based upon works supported by the Science Foundation Ireland under Grants No. SFI/02/CE1/I131 and SFI/04/BR/CS0694 and by the European Commission under the Nepomuk project FP6-027705.

<sup>3</sup> <http://en.wikipedia.org/wiki/Wiki>

Recently several Semantic Wikis have been developed, such as Platypus [22], WikSAR [2], Semantic MediaWiki [23] and IkeWiki [20]. These Wikis answer these questions in a rather limited way: (a) they allow only simple annotations of the current Wiki page; (b) they do not formally separate the page and the concept that it describes; and (c) they do not fully exploit the semantic annotations for improved navigation.

In this paper we specifically address these three questions in a broader way: in Sect. 2 we analyse Wiki annotations from a conceptual level, discuss representation mechanisms, and current annotation support in Semantic Wikis. In Sect. 3 we offer an improved navigational model based on semantic annotation; the navigation model is similar to e.g. Longwell<sup>4</sup> for faceted browsing of semistructured data, but works, in contrast to existing approaches, for arbitrary datasets with arbitrary structure. We report on our prototype implementation Semper-Wiki [12] in Sect. 4; the implementation has been updated to include these new ideas.

## 2 Annotations

In the following section we discuss our first question: how to annotate Wiki content?

Let us first analyse what an annotation is. We annotate data all the time: when we read a paragraph, and mark “great!” in the margin, that is an annotation; when our text editor underlines a misspelled word, that is also an annotation. Annotations add some information to some other information; to annotate means “to make notes or comments” [16].

Another way to view annotations is metaphorically: URIs<sup>5</sup> are the “atoms” of the Semantic Web and semantic annotations are the “molecules”. The Semantic Web is about shared terminology, achieved through consistent use of URIs. Annotations create a relationship between URIs and build up a network of data.

### 2.1 Conceptual model

We now explore the conceptual model behind annotation in more depth. The term “annotation” can denote both the process of annotating and the result of that process [9]. Where we say “annotation” we mean the result. An annotation attaches some data to some other data. An annotation establishes, within some context, a (typed) relation between the annotated data and the annotating data.

Investigating the nature of annotation further, we can model it as a quadruple:

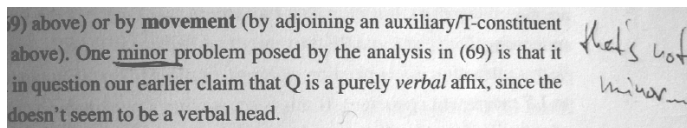
**Definition 1 (Annotation).** *An annotation  $A$  is a tuple  $(a_s, a_p, a_o, a_c)$ , where  $a_s$  is the subject of the annotation, the annotated data,  $a_o$  is the object of the annotation, the annotating data,  $a_p$  is the predicate, the annotation relation, that*

<sup>4</sup> <http://simile.mit.edu/longwell/>

<sup>5</sup> <http://www.w3.org/Addressing/>

defines the type of relationship between  $a_s$  and  $a_o$ , and  $a_c$  is the context in which the annotation is made.

*Example 1 (Informal annotation).*



The *annotation subject* can be formal or informal. For example, when we put a note in the margin of a paragraph, the informal convention is that the note applies to the paragraph, but that pointer is not formally defined. If we however use a formal pointer such as a URI<sup>6</sup> to point to the paragraph then the subject is formally specified.

The *annotation predicate* can be formal or informal. For example, when we put a note in the margin, the relation is not formally defined, but we may informally derive from the context that that the note is a comment, a change-request, an approval or disapproval, etc. If we use a formal pointer to an ontological term that indicates the relation (e.g. `dc:comment`) then the predicate is formally defined.

The *annotation object* can be formal or informal. If an object is formal we can distinguish different levels of formality: textual, structural, or ontological. For example, then string “This is great!” is a textual object. A budget calculation table in the margin of a project proposal is a structural object. And an annotation object that is not only explicitly structured but also uses ontological terms<sup>7</sup> is an ontological object.

The *annotation context* can be formal or informal. Context can could indicate when the annotation was made and by whom (provenance), or within what scope the annotation is deemed valid, for example in a temporal scope (it is only valid in 2006) or in a spatial scope (it is only valid in Western Europe). Usually context is given informally and implicitly. If we use a formal pointer such as a URI then the context is formally defined.

Combining the levels of annotation subject, predicate, and object, we can distinguish three layers in annotations: i) informal annotations, ii) formal annotations (that have formally defined constituents and are thus machine-readable),

<sup>6</sup> One can use XPointer to point to a paragraph in a document and XPointer can be used as a URI, as discussed in <http://www.w3.org/TR/xptr-framework/#escaping>.

<sup>7</sup> Ontological means that the terminology has a commonly understood meaning that corresponds to an shared conceptualisation called ontology [8]. Whether a term is *ontological* is a social matter and not a technical or formal matter. It is sometimes mistakenly understood that using a formal ontology language makes terms ontological. An ontology however denotes a shared (social) understanding; the ontology language can be used to formally capture that understanding, but does not preclude reaching an understanding in the first place.

and iii) semantic annotations (that have formally defined constituents and use only ontological terms). We have given some simple examples for each kind of annotation in Examples 1 (a handwritten margin annotation in a book), 2 (formally expressed in N3<sup>8</sup>) and 3 (formally expressed and using ontological terms), respectively. All three examples are here given without any explicit context.

**Definition 2 (Formal annotation).** *A formal annotation  $A_f$  is an annotation  $A$ , where the subject  $a_s$  is a URI, the predicate  $a_p$  is a URI, the object  $a_o$  is a URI or a formal literal, and the context  $a_c$  is a URI.*

*Example 2 (Formal annotation).*

```
<http://papers.org/minimalism#minor>
<http://localhost/schema#disagree>
  "that's not minor!".
```

**Definition 3 (Semantic annotation).** *A semantic annotation  $A_s$  is a formal annotation  $A_f$ , where the predicate  $a_p$  and the context  $a_c$  is an ontological term, and the object  $a_o$  conforms<sup>9</sup> to an ontological definition of  $a_p$ .*

*Example 3 (Semantic annotation).*

```
<http://papers.org/minimalism#minor>
  ibis:con
  [ rdf:type ibis:Argument;
    rdf:label "that's not minor!" ].
```

## 2.2 Annotations in Wikis

We can, similarly to [18], distinguish three levels of annotations in a Semantic Wiki:

**Layout** Annotations that describe textual formatting without additional structural information, such as **bold** or *italic* words<sup>10</sup>.

**Structure** Annotations that describe the structure of a page or of a set of pages, such as hyperlinks (inter-page structure), headings, subheadings, and paragraphs (internal page structure), and itemised and numbered lists.

**Semantics** Annotations that relate pages or page elements to arbitrary resources through typed ontological relations, such as categorising a page in a taxonomy, specifying the friends of a described person, or the books of a described author.

<sup>8</sup> <http://www.w3.org/DesignIssues/Notation3.html>

<sup>9</sup> The notion of “conformance” is rather weak in some ontology languages (such as RDFS or OWL) since these are not constraint-based languages (as opposed to e.g. database schemas). However, we use the notion of conformance to differ between “good” usage of textual objects, for example to indicate the name of a person, and “bad” usage of textual objects, for example to indicate the friends of a person.

<sup>10</sup> These annotations could formally be considered semantical, because they have an explicit and shared meaning, which is used by the rendering engine.

Annotations in a regular Wiki are limited to layout and structural annotations. Semantic annotations are unique to Semantic Wikis, and are the further focus of this section.

We now present one possible annotation syntax for semantic annotations, namely the one used in SemperWiki [12]. To simplify the annotations, we only consider annotations that have the page on which they appear as subject. The annotation subject is thus implicitly defined. We also limit ourselves, for simplicity, to annotations with an implicit context. The annotations are then restricted to defining the *predicate* and *object*, which is done by simply stating the two on a separate line.

The example page shown in Fig. 1 describes the World Wide Web Consortium. The page includes some English text, and some annotations which state (using the Wordnet and Semantic Web Research Community ontologies) that the W3C is an organisation lead by Tim Berners-Lee. The syntax includes referencing using namespace abbreviations, internal Wiki pages, and full URIs; see [12] for more information.

W3C
The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential.
<code>rdf:type wordnet:Organization</code>
<code>swrc:head http://www.w3.org/People/Berners-Lee/card#i</code>
<code>dc:date "2006/01/01"</code>

Fig. 1: Simple Wiki page about the W3 Consortium

### 2.3 Representation

Having defined annotations in Wikis, we now answer the second question: how to formally represent Wiki content?

RDF<sup>11</sup> is a straightforward way to represent these annotations formally, since it has exactly the same model as our annotations. We can either use standard RDF to represent annotations without context, or RDF quads (which is a common RDF extension) for annotations with context.

RDF does pose some constraints on the constituents of triples: the subject must be a URI or a blank node (not a literal), and the predicate must be a URI (not a literal or blank node). If we follow these restrictions in our annotations, RDF offers a good representation model.

<sup>11</sup> <http://www.w3.org/RDF/>

We represent pages and their annotation in RDF as follows: each page is an RDF resource, and each annotation a property of that resource. We can represent not only the semantic annotations in RDF but the whole Wiki content. The (natural language) Wiki content is captured through the predicate `semper:content`, the outgoing links to other pages through the predicate `semper:links`. Figure 2 shows the RDF graph that represents the page in Fig. 1.

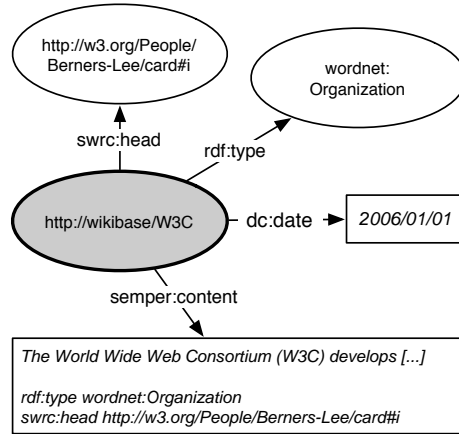


Fig. 2: RDF graph for the W3C page in Fig. 1

**Problem: documents vs. concepts** Because annotations can describe concepts (the W3 consortium) and web documents (the page about the W3 Consortium), the question arises which URI to use as the annotation subject.

For example, the Wiki page in Fig. 1 also contains the statement that it was created on January 1, 2006. But does this statement say that the *document* was created in 2006 or that the *concept* of the document, i.e. the W3C, was created in 2006? We may derive with some background information that we mean the first, but we actually need a way to say both: we sometimes want to make statements about a concept and sometimes about the document describing that concept.

This issue (often referred to as the “URI crisis”) is well-known from early discussions on Web architecture, and has gained renewed interest in the Semantic Web community. The problem is that it is unclear what a URI denotes (at least, it is unclear for URIs that are URLs, but the discussion focuses primarily on http URIs which are indeed URLs). A URL can denote a name, an abstract concept, a web location, or a document [5]. The root of the problem is that the same URI can be used to identify a subject directly (web document) or indirectly (concept that is subject of document) [15].

Hawke [10] suggests<sup>12</sup> to disambiguate the concept and the document syntactically by using the # symbol: `http://google.com/` would denote the web document and `http://google.com/#` would denote the concept. The solution is not ideal [15] since the hash symbol is a legal URI character and can be used to denote a document fragment, while referring to document fragments with URI fragment identifiers is crucial for fine-grained document annotation<sup>13</sup>.

**Solution: locators vs. names** As Pepper remarks, “using a locator for something that does not have a location is asking for trouble” [15]. The obvious solution is to not use a locator (URL) but a non-addressable identifier<sup>14</sup> (URN) for non-locatable things such as concepts.

Unfortunately, using a URN to identify concepts violates the fundamental Web principle that a URI should point to a location with useful information about the thing it identifies [4]. However, that could be remedied by using a syntactical convention (*mirror-URIs*) to relate the document URL to the concept URN, such as prefixing the URL with the `urn:` protocol handler.

To complete this solution, we need to extend our Wiki syntax in two ways to include a way:

1. to distinguish annotations about a document (Wiki page) from annotations about the concept, which we do by prefixing the annotation with the ! symbol.
2. to relate a page to the concept it describes (in case the page describes a concept in a different naming authority, e.g. a page on `http://wikibase/W3C` that describes `urn://w3.org`), which we do with `semper:about`.

Figure 3a shows how these extensions are used to now correctly state that the W3C (identified by `urn://w3.org`) is an organisation headed by Tim Berners-Lee, and that this page (identified by `http://wikibase/W3C`) was created on January 1st, 2006, and Fig. 3b shows the corresponding RDF graph.

## 2.4 Annotation in current Semantic Wikis

Having answered the first two questions (how to annotate and how to represent Wiki content), we now characterise the annotation and representation in several existing Semantic Wikis.

Annotations in Semantic Wikis are formal and possibly semantic, i.e. they are formally defined, and possibly use ontological terms. We have selected several dimensions to classify annotations in Semantic Wikis from the literature (we again focus on the annotation result, not the annotation process). We have added one new dimension to capture the important notion of annotation *context*:

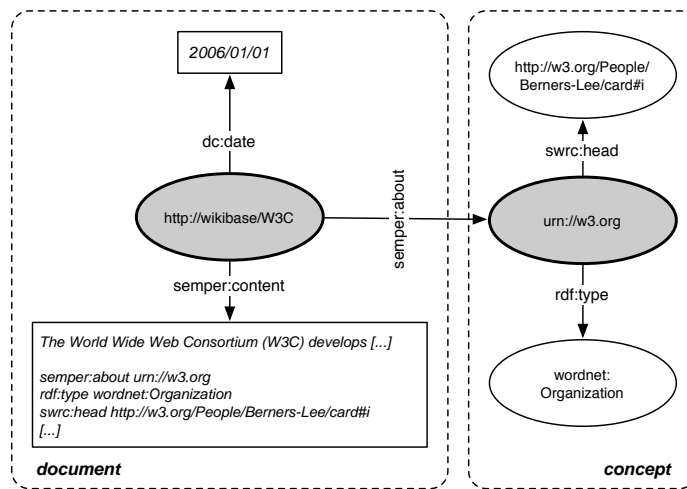
<sup>12</sup> The proposal is a bit more intricate, but for our purposes this explanation suffices.

<sup>13</sup> see e.g. `http://w3.org/TR/annotor` [9].

<sup>14</sup> Clarification on the relation between URIs, URLs and URNs can be found at `http://www.w3.org/TR/uri-clarification/`.

<p><b>W3C</b></p> <p>The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential</p> <pre>semper:about urn://w3.org rdf:type wordnet:Organization swrc:head http://www.w3.org/People/Berners-Lee/card#i</pre> <p>Now we have an annotation about the page itself: !dc:date "2006/01/01"</p>
---

(a) example page



(b) RDF representation

Fig. 3: RDF representation of an example page



- Subject attribution** (also called “scope” [19]) Indicates the subject of the annotation: is the subject of the annotation the same as the page on which it appears or an arbitrary page? In a Wiki, the possible attributions are: the page on which an annotation appears, an arbitrary page, or an anonymous resource.
- Subject granularity** (also called “lexical span” [18]) Indicates the granularity of the annotation subject: e.g. is the annotation about a document, a section inside a document, a sentence, or a word?
- Representation distinction** (also called “instance identification vs. reference” [3]) Indicates whether the Wiki distinguishes annotations about the Wiki page itself from annotations of the concept described on the page?
- Terminology reuse** (also called “interoperability” [19]) Indicates whether an annotation is self-confined with its own terminology, or whether an annotation uses terms from existing ontologies, and are thus interoperable and understandable for others.
- Object type** (also called “annotation form” [7]) Indicates the type of annotation object: is it a literal or textual object, a structural object (including a hyperlink to another page), or an ontological object?
- Context** Indicates the context of the annotation: when was it made, by whom (provenance), and within what scope: the annotation could for example be temporally scoped (it is only valid in 2006) or spatially scoped (it is only valid in Western Europe).

These dimensions can indicate the level of annotation in current Semantic Wiki approaches. We do not provide an exhaustive evaluation, but evaluate WikSAR [2], Semantic MediaWiki [23], IkeWiki [20] and SemperWiki [12] as the most prominent systems under ongoing development.

dimension	WikSAR	Sem. MediaWiki	IkeWiki	SemperWiki
attribution	current	current	current	current, any URI
granularity	page	page	page	page, any fragment
repr. distinction	no	no	yes	yes
terminology reuse	no	no	yes	yes
object type	literal, page	literal, page	literal, page	literal, page, URI
context	no	no	no	no

Table 1: Annotations in current Semantic Wikis

**Subject attribution** Most existing Wikis only allow statements about the current page. The subject of an annotation is never explicitly stated, but always implicitly assumed to be the page on which the statement appears. In SemperWiki the user can explicitly state the subject of the annotations, because we separate the page and the thing it describes (as explained in Sect. 2.3), and annotations can thus be attributed to arbitrary URIs.

**Subject granularity** Most existing Wikis only allow annotation of complete pages, not of subsections or arbitrary parts of text, for the same reason (implicitly) as mentioned above.

Since SemperWiki allows users to attribute annotations to arbitrary URIs one could annotate a document fragment as follows: create a Wiki page, point it to the document fragment using an XPointer URI, and annotate the page.

**Representation distinction** Of the discussed Wikis only SemperWiki clearly separates the page from the concept that it describes, and offers a syntax that distinguishes annotations of the page from annotations of the concept. IkeWiki also separates pages from the concepts that they describe (a concept can be represented on multiple pages), but does not, as far as we know, offer a syntax to manually express this distinction.

**Terminology reuse** IkeWiki and SemperWiki allow existing terminology to be reused in annotations (through namespace definitions or full URIs), the rest can only create annotations using internal Wiki pages and can thus not make use of existing terminology.

**Object type** All discussed Wikis allow an object to be a literal or an internal Wiki page. Of the discussed Wikis, only SemperWiki allows the object of an annotation to be an arbitrary URI. No Semantic Wiki allows unnamed resources (blank nodes) as objects.

**Context** Is ignored in all existing Wikis.

Summarising, we have developed a conceptual model for annotations in general, and for semantic annotations in the context of Semantic Wikis specifically. Given this model we have seen that current Semantic Wikis offer only limited annotation possibilities (which is not necessarily wrong, but has now been recognised explicitly), and do not clearly separate the page from the concept that it describes. We have shown how SemperWiki addresses these limitations.

### 3 Navigation

Having answered the first two questions, we now investigate the third question: how to navigate Wiki content?

When navigating an ordinary Wiki, all content is considered either a hyperlink or some natural language text. The hyperlinks between pages can be followed, and the full-text can be searched by keyword. But if users can not exactly formulate their information need, an exploration technique is necessary that helps users to discover data [11].

In our opinion, navigating a Wiki has two phases: looking *for* a page, and looking *at* a page. In an ordinary Wiki, exploration in both phases is limited to predefined hyperlinks. In Semantic Wikis, the semantic annotations structure the Wiki content, and we can use that structure to offer improved exploration through a technique called faceted browsing [24].

Existing approaches for faceted browsing rely on manually constructing the facets for a fixed data structure. But since Wiki content can form arbitrary and

fluent structures (because users can add arbitrary annotations to pages), we need to adjust faceted browsing to arbitrary data structures.

In this section, we present our approach to automatically construct facets for an arbitrary semi-structured dataset, independent of its structure.

### 3.1 Background

Faceted browsing is a superior exploration technique for large structured datasets [24,21,6] based on the theory of facet analysis [17].

In faceted browsing, the information space is partitioned using orthogonal conceptual dimensions of the data (these dimensions are called facets). Each facet has multiple restriction values; users select a restriction value to constrain relevant items in the information space.

In the Semantic Wiki, a facet corresponds to an annotation predicate  $a_p$  and a restriction value corresponds to an annotation object  $a_o$ . The annotation subject is the result (or purpose) of the faceted browsing: faceted browsing is a search process that takes the predicate and object values as input and returns possible matching the subject.

For example, a collection of art works can consist of facets (predicates) such as type of work, time periods, artist names and geographical locations. Users can select a certain restriction value (object) such as the 20th century to constrain the visible collection to only some art works. Multiple constraints are applied conjunctively.

Existing approaches [24,11] cannot navigate arbitrary datasets: they are limited to manually defined facets over predefined data structures. A technique for automatic classification of new data under existing facets has been developed [6], but requires a predefined training set of data and facets, and only works for textual data. A technique for automatic facet construction based on lexical dispersion has been developed [1], but is also limited to textual data.

### 3.2 Automatic facet extraction

We combine several existing techniques to offer faceted browsing for arbitrarily structured data. Setting up faceted browsing for a specific dataset involves two steps: i) selecting proper facets and ii) partitioning each facet into a number of restriction values.

In most existing faceted browsers, both steps are done manually: an administrator examines the dataset (e.g. a museum collection), selects useful facets (e.g. time period, artist name, location), and partitions each facet into useful restriction values: e.g. the time facet would be divided in 20 centuries, the artist facet into 26 starting letters, and the location (hierarchically) into continent and then countries.

We focus on automation of the first step: selecting proper facets.

### 3.3 Facet selection

A facet should only represent one important characteristic of the classified entity [17]. This entity corresponds to our notion of RDF resource. In RDF, each resource is defined by one or more predicates; these predicates could be considered as entity characteristics. Our goal is to find, among all available predicates, those that best represent the dataset.

**Frequency** A good predicate has a high occurrence frequency inside the collection. The more distinct resources a predicate covers, the more useful it is in dividing the information space [6]

**Distinguishing power** A good predicate has a uniform value distribution (its distinguishing power is high). A division in which the information is distributed uniformly across all partitions enables the fastest navigation to an item of interest.

**Object values** A good predicate has a limited number of different object values (between 2 and 20). If there are too many different objects to choose from, then the options are difficult to display and may disturb the user.

**Intuition** A good predicate reflects the scope of the information space and is intuitive for the user. For example, a user who only knows the author of some book will try to find it by using the facet “author”. Conversely, a user who only knows the title of a book will try to find it using the “title”.

We define three metrics (for the first three properties) that rank the appropriateness of each predicate; we exclude the mathematical treatment for brevity. Fig. 4 shows these metrics for a sample (CiteSeer) dataset. We cannot define a metric for intuition, since we cannot properly define intuition.

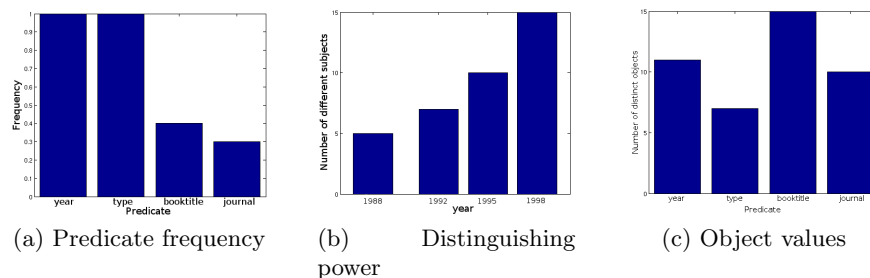


Fig. 4: Metrics in sample data

**Frequency** To measure the frequency of a predicate, we use a simple function based on the number of distinct resources that have the predicate. For example, in Fig. 4a we see that *year* and *type* occur frequently in the sample data.

**Distinguishing power** To measure the distinguishing power of a predicate we use a simple function based on the number of distinct subjects having the same object. If each object has the same number of distinct subjects, the score of the predicate is highest. For example, in Fig. 4b we see that the predicate *year* is not very balanced: there are more publications in later years.

**Object values** For displaying and usability purposes (the user should be able to have an overview of options and decide on a restriction value), the number of different object values should be approximately between [2, 20]. For example, in Fig. 4c we see that the predicate *booktitle* has many different object values, and the predicate *type* only a few (so the latter one would be more usable).

## 4 Implementation

This section presents our prototype implementations of the previous ideas.

Our open-source prototype SemperWiki<sup>15</sup> [12] was initially developed as personal Wiki for knowledge management, and therefore designed as a desktop application. The original version of SemperWiki, shown in Fig. 5, is implemented in Ruby<sup>16</sup>, using the GTK<sup>17</sup> graphical toolkit.

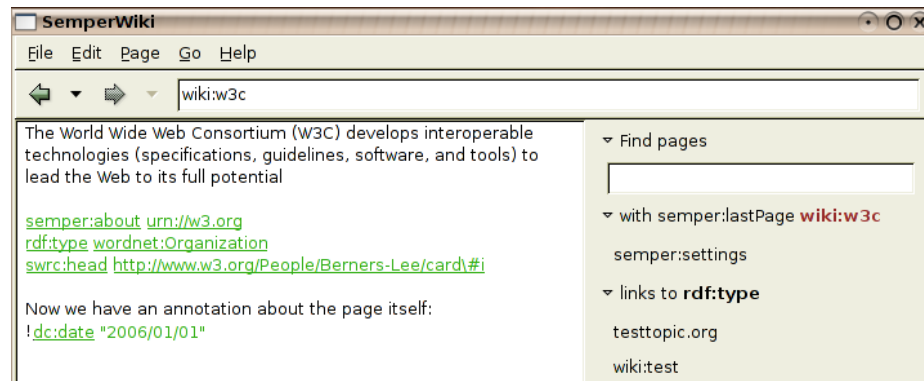


Fig. 5: SemperWiki prototype

We are currently porting SemperWiki to a Web architecture to make it cross-platform accessible, using ActiveRDF [14] and Ruby on Rails<sup>18</sup>. The new version of SemperWiki contains all the annotation functionality described in Sect. 2, and clearly distinguishes between documents and concepts, as discussed in Sect. 2.3.

<sup>15</sup> <http://semperwiki.org>

<sup>16</sup> <http://ruby-lang.org/>.

<sup>17</sup> <http://gtk.org>.

<sup>18</sup> <http://rubyonrails.org>

Secondly, we have built a prototype that implements the automatic selection of facets. The resulting faceted browsing interface is shown in Fig. 6; please note that this interface is automatically generated for arbitrary data. In this dataset, year, type, booktitle and journal are the facets (selected from the predicates), and 1988, 1992, etc. are the facet values (annotation objects without clustering). The prototype is implemented in Ruby and ActiveRDF, and works on arbitrary RDF data sources through the generic RDF API of ActiveRDF.

We have not yet done a comprehensive assessment, but an initial evaluation<sup>19</sup> looks promising: the metrics automatically select the most important predicates (such as year, type and author) as the most important facets.

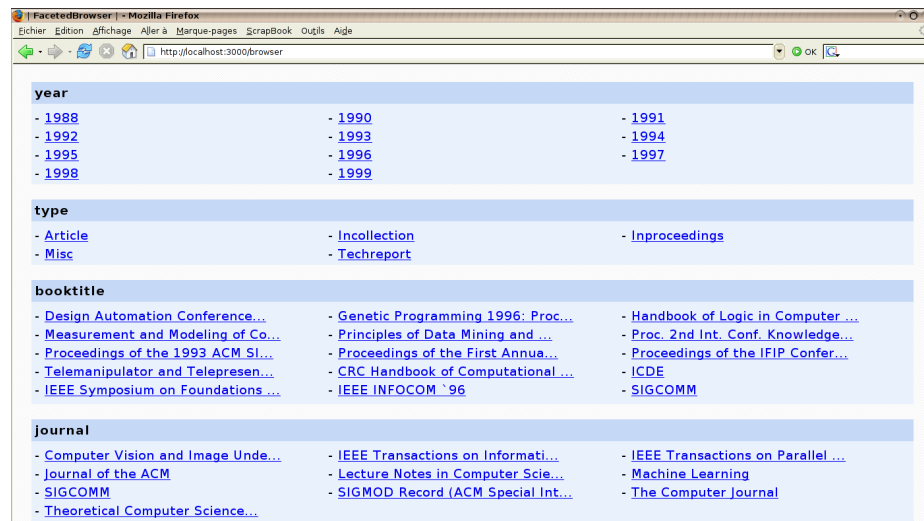


Fig. 6: Faceted browsing prototype

## 5 Discussion

The results of our work allows us to give good answers to the three initial research questions of this paper. We are satisfied with this overall results but we will also have in the following a short discussion about possible unsettled points.

Our approach for annotation in the Semantic Wiki ignores the context of annotations. Actually, to our knowledge, all annotation approaches ignore the notion of context. More research is needed on identifying and on modelling context of annotations.

<sup>19</sup> On a sample CiteSeer dataset from <http://www.csd.abdn.ac.uk/~ggrimnes/swdataset.php>.

Secondly, when annotating Wiki concepts we might encounter a naming ambiguity if two people use different URNs for the same real-world concept. But a large-scale social system as Wikipedia shows us that naming ambiguity tends to resolve over time (people reuse socially accepted names), especially if enhanced with a popularity-based recommendation system.

The solution for the representation problem of documents vs. pages, as presented in Sect. 2.3, has one drawback concerning existing RDF data. Unfortunately the world is already full of RDF statements that do not clearly distinguish documents and pages, but use URLs to refer to both. Employing our solution, encountering a URN as subject we would know that the concept is meant, but encountering a URL we would not be sure that the document is meant; the URL could be a “legacy” URL that does not conform to our distinction and is (wrongly) used to identify a concept. Our solution has therefore only limited applicability, but that is unfortunately the nature of the problem.

## 6 Conclusion

As explained in the introduction, a Semantic Wiki needs to address three questions:

1. how to annotate content?
2. how to formally represent content?
3. how to navigate content?

We have developed an elaborate model of annotations and shown how SemperWiki –as opposed to other Semantic Wikis– supports very rich annotations. We have shown how to formally represent content, and shown how SemperWiki –as opposed to other Semantic Wikis– correctly distinguishes between documents and concepts, without limiting the possible annotations. Further, we have presented how the existing technique of faceted browsing can be adjusted to flexible semistructured data, by automatically constructing facets from the data. Finally, we have developed metrics for facet (predicate) selection and techniques for object clustering inside each facet.

Faceted browsing is a superior data exploration technique [24]. We have shown how this technique can be employed for semistructured Wiki content. The technique works for any formal annotation, without conforming to a fixed data-schema; and it additionally rewards semantical annotations (because consistent use of shared terminology reduces the search space).

We are currently extending our work in several directions. First, we are integrating the faceted browser into the Web version of SemperWiki. Secondly, we are developing the clustering step of the faceted browser, and evaluating the quality of the facet construction algorithm. Thirdly, we are working on a page recommendation system, that works in the second phase of Wiki navigation and recommends (similar or related) pages to the current page, based on the structure of the Wiki content.

## References

1. P. Anick and S. Tipirneni. Interactive document retrieval using faceted terminological feedback. In *HICSS*. 1999.
2. D. Aumueller. Semantic authoring and retrieval within a wiki. In *ESWC*. 2005.
3. S. Bechhofer, *et al.* The semantics of semantic annotation. In *ODBASE*. 2002.
4. T. Berners-Lee. Putting the Web back in Semantic Web, 2005. Keynote presentation at ISWC 2005, <http://www.w3.org/2005/Talks/1110-iswc-tbl/>.
5. D. Booth. Four uses of a URL: Name, concept, web location, and document instance. [http://www.w3.org/2002/11/dbooth-names/dbooth-names\\_clean.htm](http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm).
6. W. Dakka, P. Ipeirotis, and K. Wood. Automatic construction of multifaceted browsing interfaces. In *CIKM*. 2005.
7. J. Euzenat. Eight Questions about Semantic Web Annotations. *IEEE Intelligent Systems*, 17(2):55–62, Mar/Apr 2002.
8. T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.
9. S. Handschuh. *Creating Ontology-based Metadata by Annotation for the Semantic Web*. Ph.D. thesis, University of Karlsruhe, 2005.
10. S. Hawke. Disambiguating RDF identifiers, 2002. <http://www.w3.org/2002/12/rdf-identifiers/>.
11. E. Hyvönen, S. Saarela, and K. Viljanen. Ontogator: Combining view- and ontology-based search with semantic browsing. In *Proceedings of XML Finland*. 2003.
12. E. Oren. SemperWiki: a semantic personal Wiki. In *SemDesk*. 2005.
13. E. Oren, J. G. Breslin, and S. Decker. How semantics make better wikis. In *WWW*. 2006. Poster.
14. E. Oren and R. Delbru. ActiveRDF: Object-oriented RDF in Ruby. In *Scripting for Semantic Web (ESWC)*. 2006.
15. S. Pepper and S. Schwab. Curing the web’s identity crisis. <http://www.ontopia.net/topicmaps/materials/identitycrisis.html>.
16. N. Porter, (ed.) *Webster’s Revised Unabridged Dictionary*. 1913 edn.
17. S. R. Ranganathan. *Elements of library classification*. Bombay: Asia Publishing House, 1962.
18. F. Rinaldi *et al.* Multilayer annotations in Parmenides. In *Proc. of the K-CAP2003 workshop on Knowledge Markup and Semantic Annotation*. 2003.
19. P. Sazedj and H. S. Pinto. Time to evaluate: Targeting annotation tools. In *Proc. of Knowledge Markup and Semantic Annotation at ISWC 2005*. 2005.
20. S. Schaffert, A. Gruber, and R. Westenthaler. A semantic wiki for collaborative knowledge formation. In *Semantics 2005*. 2005.
21. V. Sinha and D. Karger. Magnet: Supporting navigation in semistructured data environments. In *SIGMOD*. 2005.
22. R. Tazzoli, P. Castagna, and S. E. Campanini. Towards a semantic wiki wiki web. In *ISWC*. 2004.
23. M. Völkel, *et al.* Semantic wikipedia. In *WWW*. 2006.
24. K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *CHI*. 2003.