

Semantics on Demand: Can a Semantic Wiki Replace a Knowledge Base?

David E. Millard*¹, Christopher P. Bailey², Philip Boulain¹, Swapna Chennupati¹,
Hugh C. Davis¹, Yvonne Howard¹, Gary Wills¹

¹*School of Electronics and Computer Science, University of Southampton, UK*

²*Institute of Learning and Research Technology, University of Bristol, UK*

¹{dem, prb, cs, hcd, ymh, gbw}@ecs.soton.ac.uk, ²c.bailey@bris.ac.uk

ABSTRACT

In the same way that Wikis have become the mechanism that has enabled groups of users to collaborate on the production of hypertexts on the web, Semantic Wikis promise a future of collaboration on the production of semantically linked and ontologically structured hypertexts. In this paper we describe our efforts to convert an existing ontologically structured web site called FREMA into a Semantic Wiki specifically to enable community contribution. We compare a number of existing Semantic Wikis, and explore how the notion of *semantics-on-demand* affects a system's ability to control the creation of useful ontologies and annotations. The FREMA case study introduces a number of the problems we encountered and solved, and sets the template for others considering implementing web-based knowledge bases using Semantic Wikis. Our conclusions will contribute to the agenda for those implementing the next generation of Semantic Wikis.

Keywords: Rich Hypertext, Semantic Graphs, Semantics-on-demand.

1 INTRODUCTION

The World Wide Web is the most popular hypertext system, yet it suffers a number of problems when evaluated alongside other hypertext systems. In particular: it has a very clear separation of author and reader, which means that web users cannot change the pages they are viewing, creating web pages requires specialist skills, and collaborative authoring of a Web site is difficult.

Because the Web is a framework for distribution it is possible to create Web applications that do have these features. One general solution is a WikiWikiWeb (Wiki for short) a type of Web server (or application running on a traditional Web server) that allows any reader of its pages to alter those pages, or create new ones, by using simple web forms [12]. Crucially this allows non-specialist users to contribute to the hypertext, and enables authoring that is almost as simple as reading. This blurring of reader and author has come to be associated with Web 2.0 (or the Social Web), but is in effect a return to the aspirations of the early Hypertext Pioneers (such as Bush and Nelson) using the Web as a delivery platform [15].

The Semantic Web effort is an attempt to add machine readable semantics to the Web [3]. These semantics take the form of meta-data that can be controlled by an ontology (schemas that define the agreed classes and relationships in a domain) [7]. These annotations can be used for exchanging data

between distributed processes, but they can also be used to enhance the hypertext functionality of the Web of today, by augmenting Web content with semantics that allow the content to be reasoned about.

Additional Semantics could improve the search power of the Web. Current searches are based purely of text matches (perhaps weighted according to node importance within the hypertext network [17]). Contextual search systems attempt to improve this by looking at the context of terms to decide their meaning [4, 5]. Adding semantics to pages means that it is possible to make these distinctions explicit in the content itself and search systems can ignore homonyms but find synonyms. The DBPedia project¹ has shown how semantic information can be derived from hypertext when the structure of that hypertext is known; DBPedia scraps Wikipedia for a wide variety of information, expresses that information in RDF and makes it available on a public website for query [21].

It is also possible to go the other way and create new hypertext structures based on the semantic information. This is sometimes called *Ontological Hypertext* and usually involves the process of converting a chain of semantic relations (some of which may have been inferred) into a hypertext link or set of links [24]. This is normally done by holding the semantic relations in a knowledge base, and then converting them into hypertext structures at runtime.

Currently the ideas of the Semantic Web idea are not known to regular users, and the complexity of the task of semantically enhancing content provides a barrier to authorship. The additional complexities of navigating and viewing semantically enriched hypertexts is a continuing research field [20], as are the problems of developing and agreeing new ontologies.

Semantic Wikis are an attempt to use the Wiki concept to make semantics accessible to ordinary users in the same way as ordinary Wiki's make hypertext accessible [23]. In Semantic Wikis users are able to type pages and links, forming a semantic network that can be queried. Implementations vary, but it is possible to offer the kind of semantic search promised by the Semantic Web, and ontological hypertext in the form of inline queries that construct links on the fly from semantic information authored in the Wiki. Semantic Wikis make semantics accessible because they are inherently freeform in nature and are non-restrictive, allowing the creation of *semantics-on-demand*, without a complex ontological design process beforehand.

In this paper we describe work we have done in the FREMA project to convert a Website based on a Knowledge Base (built with the traditional KB or Semantic Web approach) into a Semantic Wiki. This involved not only converting the set of instances, annotations and relationships into Wiki pages and links, but also dealing with the loss of a controlling ontology, and the fact that new relationships, attributes and classes might be added ad-hoc by users in the future.

The original FREMA knowledge base was created throughout 2005, and the Semantic Wiki development work completed at the end of Summer 2006. The final Wiki contains around 700 pages and in the period from Sept 2006-May 2008 has been relatively popular with around 270,000 page views, although the number of edits is much lower (6,125) indicating that the majority of users browse the wiki rather than contribute to it (this may be because only registered users are permitted to contribute to protect against vandalism).

The case study is interesting because it is an example of a website that must be both collaboratively edited (because it is a community resource) and also semantically rich (because the community query the semantics to help analyse the state of the domain). As such it is not a toy example of a Semantic Wiki, but a large complex tool with real users and real requirements.

¹ DBPedia Homepage: <http://dbpedia.org/> (accessed May, 2008)

2 MOTIVATION

Our intention in this paper is to give a detailed account of our experience in converting a website based on a KB behind the scenes, into a Semantic Wiki where the KB is up front and editable by all. Specifically we intend to answer two questions:

1. What are the difficulties in *converting* between the two approaches (i.e. taking an existing knowledgebase and transforming it into Semantic Wiki pages)?
2. What are the compromises made when *running* a Semantic Wiki rather than a KB; in terms of functionality, usability and manageability, and do these outweigh the advantages of flexibility and usability that the Wiki approach brings?

The authors became involved in Semantic Wikis at the end of 2005 when they were involved in the FREMA (Framework Reference Model for Assessment) project, which aimed to define a Reference Model for the domain of e-learning assessment [13]. In this context a Reference Model is a description of a how a number of services can work together to fulfill a use case. Because Assessment (meaning student evaluation) is such a rich area we did not think that it was enough to define these services in isolation, and therefore a crucial part of the FREMA work was to create a domain definition, this is a database of projects, software, and standards in the assessment domain that can act as an evidence portfolio to demonstrate that the reference model covers the parts of the domain that the community values.

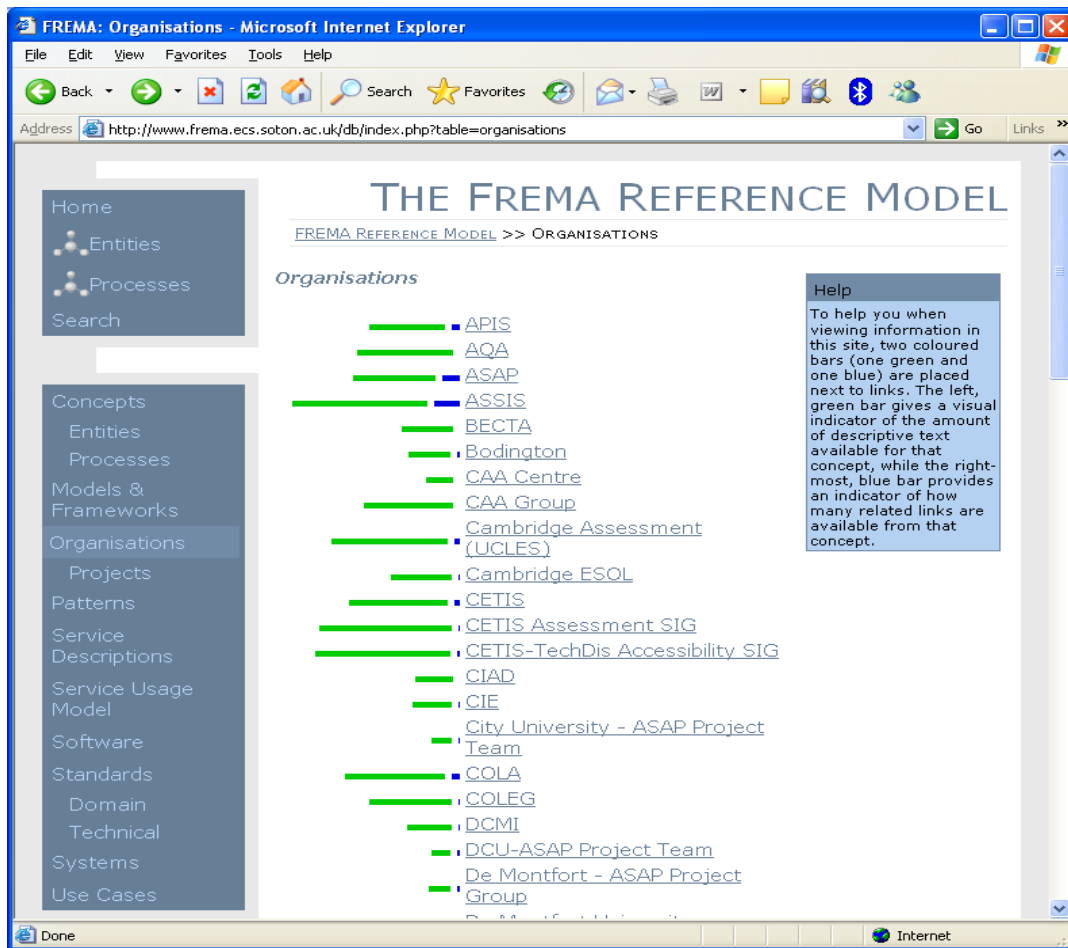


Figure 1: FREMA KB List of Organisations

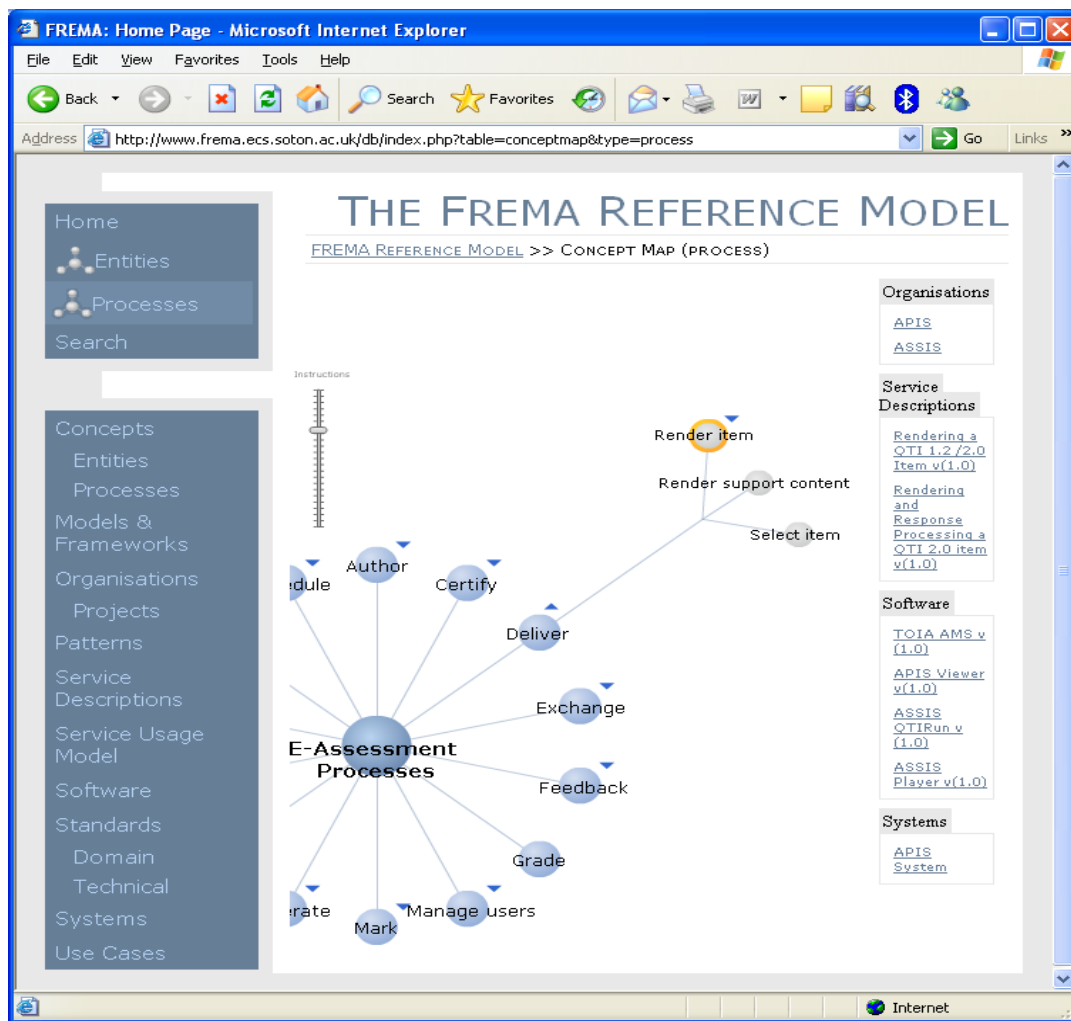


Figure 2: FREMA KB Concept Map Tool

In FREMA the domain definition was delivered as an ontologically modeled knowledgebase. Users can navigate using a concept map and then drill down to the projects, software and standards linked to any given concept [14]. The resource pages themselves are created at runtime by PHP scripts that query the knowledge base and populate a common template layout. Each resource has a description that is rendered at the top of the page, and the relationships are listed below.

Figure 1 shows the FREMA KB-based website displaying a list of items by type (Organisations). Figure 2 shows the concept map interface into the KB, when the user selects a concept the items linked to it appear in the right-hand pane. Figure 3 shows a resource page that is displayed once an item is selected; the content includes a description, external links, concepts and other relationships.

FREMA was never intended to be a static resource and we wanted the Assessment Community to be able to use the KB to record their own projects, services and potentially new reference models. We had a choice, either we developed an authoring interface to the KB that allowed users to author or alter certain aspects (such as adding new instances of classes and relationships) or we converted the whole site into a Semantic Wiki, that allowed users not only to author new instances, but also to create new classes and relationships as they saw fit.

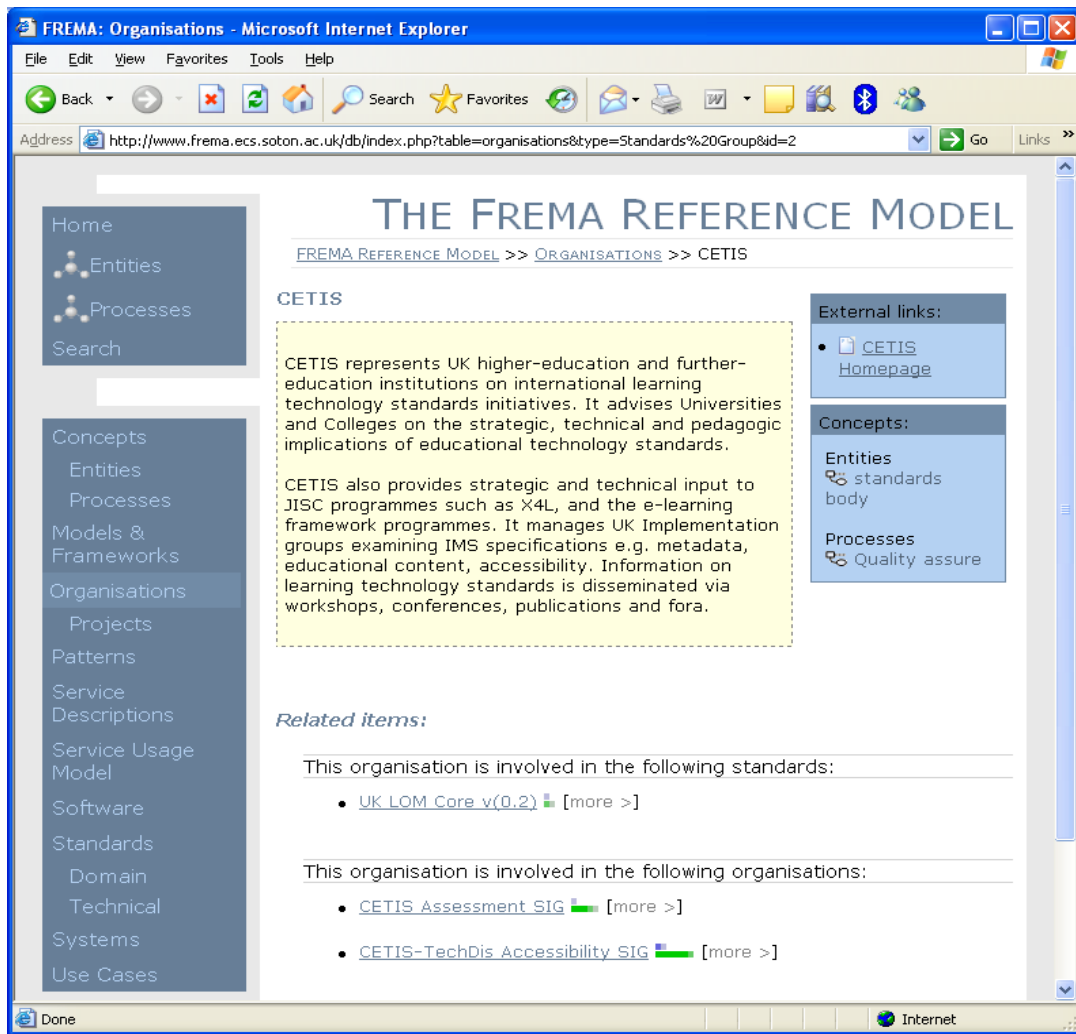


Figure 3: FREMA KB Resource Page

We chose the later approach, as we believed that it would be most likely to guarantee that the FREMA website remained fit-for-purpose as the domain evolved. We also felt confident that because the Wiki would be seeded using the information from the KB that it would start of with an appropriate structure, which could be maintained and evolved by a user community.

3 COMPARISON OF SEMANTIC WIKIS

Rather than construct our own Semantic Wiki system we wanted to leverage an existing system that had typed links, nodes, and first-class types. However, the system must not be so complicated as to require expertise in knowledge engineering to use.

Semantic Wikis are a recent topic, but are receiving increasing attention. The Semantic Wiki Community² has run a number of workshops on the subject, and there are a number of systems available. In the next section we will describe some of the major systems (although not all were available at the beginning of 2006 when we began our Semantic Wiki development work).

² Semantic Wiki Interest Group: <http://www.semwiki.org/> (last accessed May 2008)

3.1 Existing Semantic Wikis

A number of semantic web-enabled Wiki systems were considered for their fitness as a basis for converting the FREMA KB. Where possible we have referenced academic papers describing the work, but several interesting Wikis have not been formally published, in these cases we have referred to the homepage of the Wiki in a footnote.

IkeWiki is a complete rewrite of MediaWiki (the Wiki system which powers Wikipedia) in Java, with powerful semantic features [19]. IkeWiki is aimed at knowledge engineers and advanced users, and has multiple editing perspectives.

Kaukolu is a JSPWiki-based Java Wiki, which uses Sesame 2 as a store for RDF data [9]. Semantics are encoding in a N3-like plaintext format within the nodes, and there are plans to expand this to be more tolerant of natural language constructs.

Makna³ is based on JSPWiki, and uses Jena for inference. It is a sophisticated system which exposes knowledge engineering terminology (although this has been partially addressed through end-user documentation, semantic markup references, and using 'rdfs:label' to display predicates in a friendly way).

OntoWiki is a community-editable knowledge base [8]. The interface is very much technical and geared around dealing with metadata rather than content. The system has a few social web features, such as user ranking of resources, and popularity by number of viewings. Similarly to Makna, the interface is also geared towards knowledge engineers.

OpenRecord⁴ is more of a collaborative knowledge base than a Semantic Wiki, written in client-side JavaScript, with a PHP backend. Pages do not particularly have content in the conventional Wiki sense – they have a summary, and a number of sections that contain views to data in the system. Standard views include tables, lists, and graphs.

Platypus Wiki is written in Java, and licensed under the GNU GPL, it is one of the older Semantic Wikis, originating in late December 2003 [4]. Development of Platypus Wiki has been intermittent, but developers were intending to add novel features such as the ability to automatically determine the type of a page by employing Bayesian networks.

Rhizome is part of a whole set of *Rx* technologies, and is written in Python on top of the Raccoon application server [22]. A notable feature is that even Wiki information is expressed as RDF (for example, revision information).

Semantic MediaWiki (SMW) is an extension for MediaWiki which adds typed links both to nodes and literals [23]. Classes are implemented through the Wiki Category system and a transitive 'is-a' relation. SMW is implemented as an extension of MediaWiki, and thus provides widely tested Wiki features, such as discussion pages and templating.

SweetWiki identifies itself as a system that uses an ontology for the wiki and not use a wiki for an ontology [2], as a result it can describe more than one resource on the same page, but does not version knowledge structure alongside the Wiki content. The wiki is written in JSP and uses XHTML and XML for its hypertext parts and RDF, RDFS, OWL and SPARQL for semantics and querying. It

³ Makna Homepage (Nov 06): <http://www.apps.ag-nbi.de/makna>

⁴ OpenRecord Homepage (Nov 06): <http://openrecord.org>

uses a WYSIWYG editor that removes the need for wiki languages such as WikiML or CamelCase, and uses a Web 2.0 tagging metaphor for folksonomy creation.

WikSar is a simple Semantic Wiki written in Perl with a classic Wiki look and feel including CamelCase [1]. It has simple inline query support and can also display trees of relations, (e.g. using ‘IsA’ to generate a taxonomy). However, it lacks some common Wiki functions such as versioning and user identification.

Table 1: Comparison of Characteristics of Semantic Wiki Implementations.

◆ symbol implies that a system has the characteristic of that row, the ◇ symbol that it does not.

		IkeWiki	Kaukolu	Makna	Onto Wiki	Open Record	Platypus	Rhizome	Semantic Media Wiki	Sweet Wiki	Wik Sar
Key	Pages are Concepts	◆	◆	◆	◇	◇	◆	◆	◆	◇	◆
	First Class Types	◆	◇	◆	◆	◇	◆	◇	◆	◆	◆
	In-line Annotations	◇	◆	◆	◇	◇	◇	◇	◆	◇	◆
	Typed Links	◆	◇	◆	◇	◇	◇	◇	◆	◇	◆
Functional	Ontology Support	Combined	Instance first	Onto' first	Ontology first	Ontology first	Ontology first	Instance first	Instance first	Combined	Inst' first
	Inference Support	◆	◇	◆	◇	◇	◇	◇	◇	◇	◇
	In-line Queries	◆	◆	◇	◇	◆ (views)	◇	◇	◆	◇	◆
	Suggest Link Targets	◆	◆	◇	◆	◆	◇	◇	◇	◇	◇
	Suggest Link Types	◆	◆	◇	na	na	◆	◇	◇	◇	◇
	Semantics Versioned	◇	◆	◆	◆	◇	◆	◇	◆	◇	◆
	Show Inc' Relations	◆	◇	◆	◆	na	◇	◇	◇	◇	◆
Format	Semantic Markup	na	N3	Bespoke	Forms	Forms	RDF	RDF	Bespoke	Forms	Be-spoke
	Export Format	RDF	RDF	RFD / N3	RDF	none	RDF	Bespoke	RDF	RDF, RDF/A	none
	Import Format	RDF/XML	RDF	RDF / N3	none	none	RDF	Bespoke	none	RDF, RDF/A	none

3.2 Comparison

Table 1 shows a comparison of the Semantic Wikis we have described. The Table shows the state of these Wikis as they now exist, although when we first looked in 2006 their immaturity made it difficult to accurately compare them (for example we were unable to find running implementations of IkeWiki or Kaukolu at that time), and thus their functionality was taken on trust from their documentation. At that time SweetWiki had yet to be released, and so was not considered (although it is now available and we have included it in our Table for completeness).

We performed a structured comparison based on the collected characteristics of the nine systems. Where possible we verified the claims of the documentation against a running demonstrator. If a characteristic was not described, or we could not find it in the demonstrator, we assumed that it wasn't currently supported.

3.3 Summary

Although a full analysis is beyond the scope of this paper we believe that there is a type of Semantic Wiki that fuses wiki and semantic elements in such a way that the wiki mechanisms and structures are used directly for the creation of semantics. Others have made similar observations and have named this the *Wikilogy* approach [10]. Based on our experiences this type of Semantic Wiki has four characteristics:

- *Content Pages are Concepts* – although many Semantic Wikis allow users to create statements about arbitrary resources, we believe that a key characteristic of a Semantic Wiki is that pages in the Wiki become resources that can be annotated, i.e. they become concepts in the semantic graph. Such systems are equating concepts and content; often systems that do not use pages as resources instead have pages that represent *concepts*, and content is instead modeled as literal attributes of those concepts (e.g. using `rdf:label`). This later approach is more akin the way in which things are modeled on the Semantic Web, but makes authoring more cumbersome, and breaks the Wiki paradigm of directly creating the semantics as a simple byproduct of creating a hypertext.
- *First class Types/Relations* – Many of the Wikis expose the types of resources and relations as new pages in their own right. In this way it is possible to document new relations and types, which can encourage their reuse by other Wiki users.
- *In-line Annotations* – We would argue that in-line annotations are a key feature that differentiates a Semantic Wiki from a Wiki that has been coupled with a Knowledge Base. With in-line annotations the semantics are written in the same mode as the Wiki content, which seems more in the spirit of the original Wiki vision of blurring readers and authors. For example, the following is a simple page from Semantic MediaWiki that declares itself as an instance of class Project `[[Category:Project]]`, and includes a typed relationship to another instance `[[is funded by::JISC]]`. The Semantic Markup is included in the same way as the presentation markup for the Title `== FREMA Project ==`.

```
[[Category:Project]]
== FREMA Project ==
The FREMA Project was [[is funded by::JISC]].
```

Wikis that have been coupled with a KB have two separate authoring interfaces, one for the Wiki and another for the KB, and are likely to be more oriented at Knowledge Engineers

- *Typed Links* – A few of the Wikis examined allowed both traditional Wiki links and Semantic annotations to be specified in their pages, however this type of Semantic Wiki converges these two mechanisms into one, so that there is no difference between a Wiki link and a semantic relation. This does not preclude the ability to make *additional* annotations that are not links (for example, to literals, or from subjects other than the current page).

Semantic Wikis that do not demonstrate these key attributes stray further from the philosophy of the original Wiki systems. Many systems in this category effectively function as Collaborative KBs, allowing their users greater control over the knowledge structure, but at some cost to simplicity.

There are a number of other characteristics that do not seem to be so important in terms of convergence of Wikis and the Semantic Web; these differentiate the functionality of each system.

- *Ontology Support* – Semantic Wikis can utilise ontologies in a number of different ways. Some enable users to upload or define ontologies which then restrict the types that can be used within the Wiki, this is an *ontology first* approach. This can still be flexible as the system may allow users to edit the ontology within the Wiki, but it still requires new instances, annotations and attributes to conform to the current ontology. However, another common method is for the Wiki to allow users to create new classes and relation types *without* restriction, in effect the ontology can be derived by examining the instances, this is an *instances first* approach, and results in ontologies that are sometimes called Folksonomies. Constraining users via an ontology results in a well-structured hypertext that has a pre-designed structure (good for designed applications), while allowing users to define the ontology as they go results in a more chaotic space, but one which is less restrictive and thus easier to write in (good for evolving applications and evolving ontologies). Some Wikis combine these approaches, by allowing ontologies to evolve through instance declarations, but also for rules to be applied through ontological restrictions if and when they are defined.
- *Inference Support* – does the Wiki have a built in inference engine (i.e. a system of uploading ontological rules that result in new relations being inferred from those already authored in the Wiki). This inference may be built into the Wiki pages (so that inferred relations are summarised for each page) or they could manifest themselves in query results. Krötzsch et al. [11] provide a good overview of reasoning and inference in Semantic Wikis (particularly IkeWiki and SMW).
- *Inline Queries* – although all the Semantic Wikis have mechanisms that allow the semantic network to be queried, some support inline queries, which allow Wiki content to be automatically embedded in the Wiki pages on the fly.
- *Suggest Link Targets* – to aid semantic authoring, some Wikis will reflect on the authored instances and suggest potential targets for new links based on the types of the subject and relation.
- *Suggest Link Types* – also to aid semantic authoring, some Wikis will reflect on the authored instances and suggest a relation type given the types of a subject and object, both of these characteristics have the added value that they encourage the reuse of types and thus promote ontology creation.
- *Semantic Info Versioned* – Wikis tend to have sophisticated versioning systems that provide a safety net for collaborative authoring. Some Wikis extend this versioning mechanism to the semantic annotations (and thus in instance first examples, the implicit ontology). Wikis that have solely inline annotations effectively get this characteristic for free.

- *Display Incoming Relations* – Although annotations may be authored inline, many Wikis also summarise the annotations for each page, in some cases this will list not only relations for which this page is a subject, but also those for which it is an object, effectively showing incoming links.

For each of the systems we also recorded what semantic standards they used:

- *Semantic Markup* – describes the markup syntax used by Wiki authors for creating annotations and attributes (this is separate from the way in which the Wiki models these structures internally). Typically this syntax is N3 or RDF, however some systems use a bespoke method that is special to the system, and some use web forms which hide the syntax.
- *Export* – describes the export formats for the semantic annotations (if any).
- *Import* – describes the formats of annotations that can be imported into the Wiki (if any).

For the FREMA system we had a requirement that the system must be accessible for non-specialist users, which implies systems that support inline annotation authoring. We do have a pre-determined ontology, but our requirement is that this should evolve and be extended by the community, and so we also require an instance first approach to ontologies. We also have a number of analysis tools that require inline queries to embed dynamic information in the Wiki pages.

In the end we chose Semantic Media Wiki (SMW) as it is relatively mature (as it is based on MediaWiki), has a large user base, offers a number of Wiki features (such as image and user management) and fits our key criteria. It is also extensible, which allowed us to add some of the other FREMA KB tools.

4 EXPERIENCES OF CONVERSION

When we converted the FREMA KB into a Wiki we chose a simple conversion pattern, designed to replicate the look and feel of the original website. Each resource in the KB became a page on the Semantic Wiki, each relation between resources became a typed link between pages, and meta-data about resources (such as links to home pages) became typed attributes on the pages.

Because the original KB and the Semantic Wiki effectively model the same form of typed graph we expected the conversion process to be trivial, however there were a number of differences in the two approaches that made it more complex. The following points are particular to our experience with Semantic MediaWiki, but they highlight conversion problems that may be experienced with a wider set of Semantic Wikis that take a similar approach.

4.1 *IDs vs. Names*

In the KB all the instances, classes and relations have unique ids which are not intended to be human readable. These are then linked to readable names that are used for titles and lists. This is typical of the RDF approach, where a resource will be identified by a unique URI, and then an `rdf:label` relation will assign it a human readable name. When converting to Wiki pages it is necessary to lose the id, this is because in a Wiki, names and ids are intrinsically linked (the name is part of the URI).

There are two consequences for this. Firstly, scoping conventions for constructing URIs that are typical in RDF have to be abandoned in favor of the URI structure used by the Wiki implementation. For example in the original FREMA KB we used the typical URI conventions seen in RDF, so the URI for the FREMA project would include the *orgs* subpath (as FREMA is a type of organisation):

<http://www.example.org/orgs/projects#frema>

which is flattened in SMW to:

<http://www.example.org/wiki?name=frema>

This can cause problems when names are not unique, for example in the FREMA KB it is typical for projects to have the same name as the systems they create.

The Wiki URI structure exists to encourage easy linking, and because a scoping hierarchy is cumbersome to evolve. Instead of scoping, Wikis take the approach of creating Disambiguation pages, which are nexus pages that take the name in dispute and describe to human readers the difference, linking to more specifically named versions. So for example, in FREMA we model a project called APIS that also created a system called APIS, resulting in the following URIs in the original KB:

<http://www.example.org/orgs/projects#apis>

<http://www.example.org/systems#apis>

in SMW this becomes a disambiguation page:

<http://www.example.org/wiki?name=apis>

which in turn points to:

http://www.example.org/wiki?name=apis_project

http://www.example.org/wiki?name=apis_system

This way authors have the choice of lazy linking (to the disambiguation page) or more specific linking (to the correct version of the page). It also means that the Wiki can evolve in a robust manner, as forking page versions and creating disambiguation pages does not break existing links.

In conversion this difference meant an expensive semi-manual process of identifying name clashes and creating disambiguation pages. This was a consequence in our choice of Wiki as other Semantic Wiki implementations – such as IkeWiki and SweetWiki – do separate URLs from IDs.

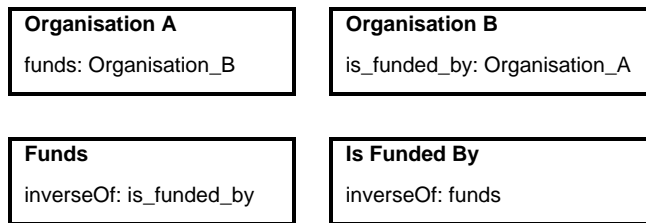
If SMW had used namespaces to differentiate between different Categories (classes), the need for disambiguation would have disappeared (it would also be useful for ontology autocompletion, see Section 4.3), however Wikis tend to use namespaces for modeling different system areas (e.g. Help, Discussion, etc), and adding semantics to the URI makes authoring more difficult, as the user often has to scope links.

4.2 *Unidirectional vs. Bidirectional Relations*

One of the most significant differences between the KB and the Wiki is the way in which relations are stored. In the FREMA KB, like RDF, relations are recorded once. When viewing a page representing a resource at either end of the relation, the relation is rendered. Unlike RDF the FREMA KB also stored a reverse name for the relation so that the relation can be described naturally in both directions (Organisation_A *funds* Organisation_B, Organisation_B *isfundedby* Organisation_A). However, in Semantic Wikis that store relations on pages, this means that either a choice must be made as to which page the relation is stored in, or it must be stored twice.

Because in the conversion process we have a clean model of the semantic network it is possible for us to store the relation twice, once on either page, where the second version represents the inverse relation. However, in doing this the knowledge that one relation is the inverse of the other is lost. To capture this

we used the OWL relation `owl:inverseOf` and recorded this relation on the two Wiki pages that represent each direction of the original relation. For example, the organisation funding triple becomes the following four pages:



This captures all the original semantics within the Wiki pages, but unfortunately because the SMW query language does not understand the `inverseOf` relation it means that unless there are two copies of every relation query results may be incomplete. In our conversion process we do make sure that two copies (one in each resource page) are created, but we cannot guarantee this will remain true once users start using the Wiki to create new content.

4.3 Forms vs. Freeform Authoring

The FREMA KB was controlled by an ontology, however that ontology was not explicitly modeled in the system (it existed in our design documentation). In the KB the ontology manifested itself in the forms used to author pages, which restricted the classes of resource that could be created, and the relationships that could be authored between them.

In SMW there is also no formal ontology, but the ontology manifests itself not in authoring restrictions, but in the existing pages and relations. Restricting the semantics that are allowed ensures that a well structured semantic graph will result, but SMW allows any relations and attributes to be authored, which introduces a danger that no consistent semantic graph will emerge. Because we are converting an existing KB we are in a good position, as we already have a well-formed semantic graph. In Wikis that use introspection of the existing graph to suggest link targets and types, the structure would be reinforced by new links, as users would be prompted to reuse the existing relations.

However SMW does not prompt users in this way. We therefore extended SMW to provide this functionality. In our version the user selects 'Assign Relations' rather than 'Save' on the edit page, this causes the Wiki to examine its semantic graph and make suggestions for each link type based to the type of the resources at each end of the link, and the frequency of link types in the graph. For example if a user links between *Service Description* and *Software* there are two relation types that might apply (*provides* and *implements*) and the system would suggest them according to the number of times that they have been used. The set of link types suggested would increase as users defined new types and classes, but initially this would reinforce the knowledge modeling that exists within the semantic graph inherited from the KB.

A problem with this approach is that an author may create a link to a resource that does not yet exist, and since a destination that does not exist has no resource type, it is impossible to recommend a relation type. Once again using namespaces would solve this problem (as we would know the type of the destination by its location in the namespace), but namespaces add a layer of complexity to the creation of new pages and are really addressing a different issue (see Section 4.1).

We get around this problem by first suggesting a type for the destination resource (by examining the existing graph for classes already related to this one). Once the user has selected a type for the destination resource, we can then suggest a relation type. However, we do not remember the destination resource type that they have selected, this is because the resource page does not exist, and there is therefore no way of recording the type of that resource. This means that while our tool prompts users to add semantic annotations, it does not guarantee consistency until the resource is actually created.

4.4 Analysis Tools vs. Generic Queries

The FREMA KB-driven website included a number of tools that analysed the KB in order to communicate to users some view of the domain being modeled. In particular there were three tools:

- *Information Bars* (on lists we displayed a small bar which showed users how much information lay behind that link)
- *Gap Analysis* (for Use Case Diagrams, we displayed a table showing which services could be used to provide this use case, and which of those services had working implementations)
- *Domain Topology* (this showed a mapping of one resource type against another, in order to show patterns across the domain. For example, by enumerating software against standards it is possible to see which standards are popular, by enumerating projects against concepts it is possible to show where interest/funding lies within the domain)

These tools were specific to the FREMA Project, but are indicative of the sort of usability tools that are normally attached to KB-based Web Systems. In the Semantic Wiki we had to replace these tools using the generic query mechanism of the Wiki. There were two factors which affected our ability to do this: the ability to count relations (needed for Information Bars, and Topology Tools), and the ability to perform complex join queries (needed for the Gap Analysis).

In SMW the query language is capable of performing one level of query joining. The full Gap Analysis requires two levels (the 'For each' statements in the following pseudocode):

```
For a given use case diagram:
  Get all the use cases
  For each use case
    Get all services that fulfill it
    For each Service
      Get all the implementations
```

We managed to get around this by breaking the Gap Analysis into two stages:

```
For a given use case diagram:
  Get all the use cases
  For each use case
    Get all services that fulfill it
```

```
For a given use case diagram
  Get services that fulfill its use cases
  For each service
```

This displays the same information, but does not map the service implementations through to the use cases themselves.

Unfortunately SMW does not support counting relations within its query mechanism. We could have extended the Wiki with special functions that provide this information, but in the end this was not possible in our projects timescale. Some Semantic Wikis support the SPARQL query interface, which would have been powerful enough for our needs, but this would have to be coupled with the inline query mechanism in Semantic MediaWiki, as most of these advanced queries would be set up by more technical users, but would be of interest to all the users of the Wiki.

4.5 Transclusions vs. Repetition

Transclusion is a hypertext term coined by Ted Nelson [16]. It refers to the process of including in a hypertext document content that is stored elsewhere. Generally on the Web, images are transcluded (one image may appear in multiple web pages), but Nelson originally imagined that everything would be transcluded from a centrally managed *permascroll*, meaning that there would be no repetition of any content.

The FREMA KB uses transclusion to build up pages about a resource. Certain resources do not make sense without the description of related resources, for example *Service Implementations* (information about specific pieces of software that implement services) are not meaningful without *Service Descriptions* (information on the abstract behavior of a service).

MediaWiki supports transclusions, but in SMW this mechanism must be used carefully as resource descriptions are not stored separately from their relations. The resource description is polluted with relation, attribute and layout information, which means that if that content is transcluded the semantic information is also transcluded; this would mean that the transcluding page inherits semantic information that is not appropriate for it. We considered the option of defining the description as an attribute of the page (as a string literal) to separate content and relations and allow the description to be transcluded without any semantics. The way that SMW supports inline attribute definitions would make this possible, but it was a cumbersome solution, which might easily break if someone who didn't understand how those definitions were used happened to edit the page. Subpages are another mechanism that might allow the kind of transclusions we wanted, but in both of these cases it adds unwanted complexity to the editing process.

We decided that we would have to replace transclusions with repetition, and allow Wiki users to evolve the content separately if they wished. This is normal Wiki practice, for example it is common on Wikipedia for part of a page to be promoted to a full page in its own right, and for the original section to become an independent summary. This creates some system overhead, as we are storing content twice, and also means that the wiki is in danger of becoming inconsistent, however we felt that the largest number of users would understand this simple mechanism.

4.6 Consistent vs. Inconsistent Presentation

Each page on the original FREMA website was generated automatically by scripts that queried the FREMA KB. This meant that each page had a consistent look-and-feel, with a single title, a section describing the resource, and then a bulleted list of related items. When we moved to the Semantic Wiki

we lost the ability to ensure consistent presentation across all the pages, as this is now left to the individual authors of each page.

It is debatable as to which should be preferred, on the one hand a common look-and-feel enables users to find information quickly across a range of pages, and ensures that the site looks consistent and coherent. On the other hand, having some flexibility in the way in which resources are presented can result in each resource being presented in the best way for that resource.

SMW encourages some common look-and-feel through the use of templates, and because our Wiki-pages were generated based on the original KB, they all have a common structure. One interesting observation is that this has actually discouraged people from contributing to the Wiki, because strict conformance to some presentation rules makes the content seem more formal, and less community owned. It is our hope that as the Wiki expands with new content this effect will be reduced.

4.7 *Concept Maps*

An important part of the FREMA KB was the use of concept maps to provide an overview of the domain. The idea was that a user who knows what they were looking for could either search, or use the lists (organized by class) to find resources, while a user who was unsure of the area could browse using the concept maps, and serendipitously find resources that were related to their topic of interest. Via a community consultation process we produced a map of *entity* types that are considered important within the domain, and another concept map of the common *processes*. Every resource in the reference model is associated with at least one concept from each map. We then built a flash front end that displayed both maps, and linked through to the KB resources.

In the FREMA KB the concept maps were static, and we were able to embed their layout in the flash file itself, however in the Semantic Wiki the concepts become pages along with every other resource, and as a result the concepts, and the structure of the concept maps, are now subject to change. To overcome this we rewrote the concept maps to initialise themselves based on RDF exported from the Semantic Wiki. SMW has a simple recursive mechanism to export relations based on a starting resource, so it was easy to access the concept information by exporting the RDF about the central node of the concept map. To avoid exporting the entire semantic graph of the Wiki the export function limits itself to two recursions (two annotations traversed from the starting resource), this meant that the export RDF contained only the top three levels of the concept graph. It is possible to turn off this limit and recursively export all relations, but this produces an RDF file that is prohibitively large.

We got around this problem by accepting the three level limit and directly querying SMW's backend for any additional information we needed (about what resources belong to what concept) as and when the user clicks on the concept graph. Ideally SMW would be extended to include a filtered export function, that recursively returned all nodes and relations within a restricted set of types.

A larger problem was that users of the Wiki have reported that while the concept maps make a lot of sense when presented as a graphical map (through the Flash front end), concepts are difficult to understand when encountered as pages in the Wiki (with typed links to parents and sub-concepts). Our intention was that normal users would only access the concept map through the graphical front end, and that more specialist users would edit the concept map through the Wiki pages, but there is nothing to stop any user following a link to a concept page, and once there the page has the appearance as the rest of Wiki. As a result the impression of the concept-maps as a high level view of the domain is lost, and the concepts appear to be part of the domain itself.

This was not so serious as to prevent users from using the Wiki or the Concept Maps, but it complicates the system, when simplification was the goal.

5 MODIFICATIONS AND IMPLEMENTATION

Converting the FREMA KB into a Semantic Wiki was not the trivial process that we hoped for, even after carefully choosing a more mature Semantic Wiki. However, by sacrificing some of the functionality of the original site, and writing limited SMW extensions, it was possible to replicate most of the original website (and we gained all the advantages of using a Wiki: open editing, administration, discussion, file and image management, etc). We sacrificed the following functionality:

- Unique IDs – became unique names / disambiguation pages
- Bi-directional relations – became two uni-directional relations and an *inverseOf* relation
- Transclusions – were replaced with repetition
- Information Bars – were removed
- Domain Topologies – were removed
- Concept Maps – became limited to three levels

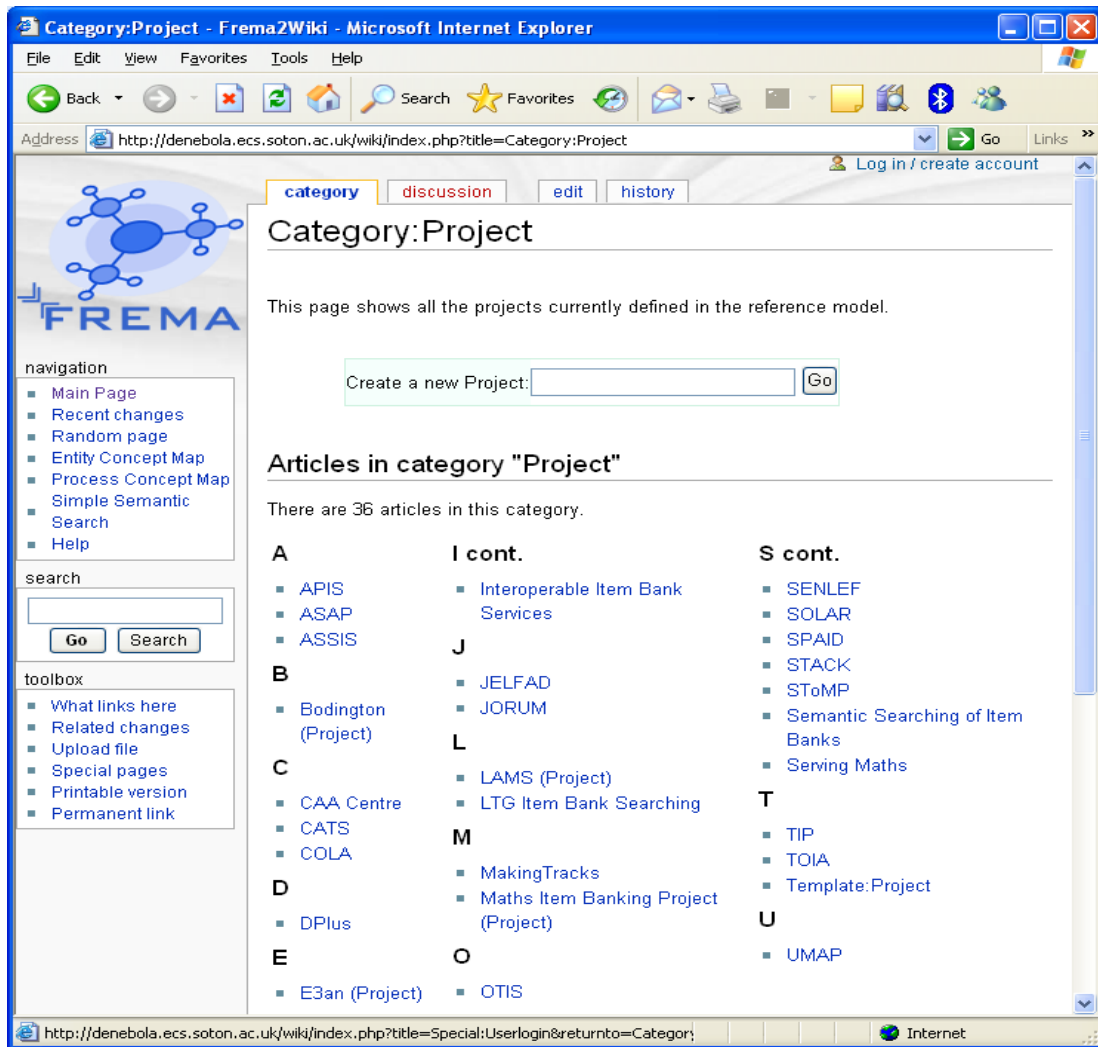


Figure 4: FREMA Semantic Wiki List of Organisations

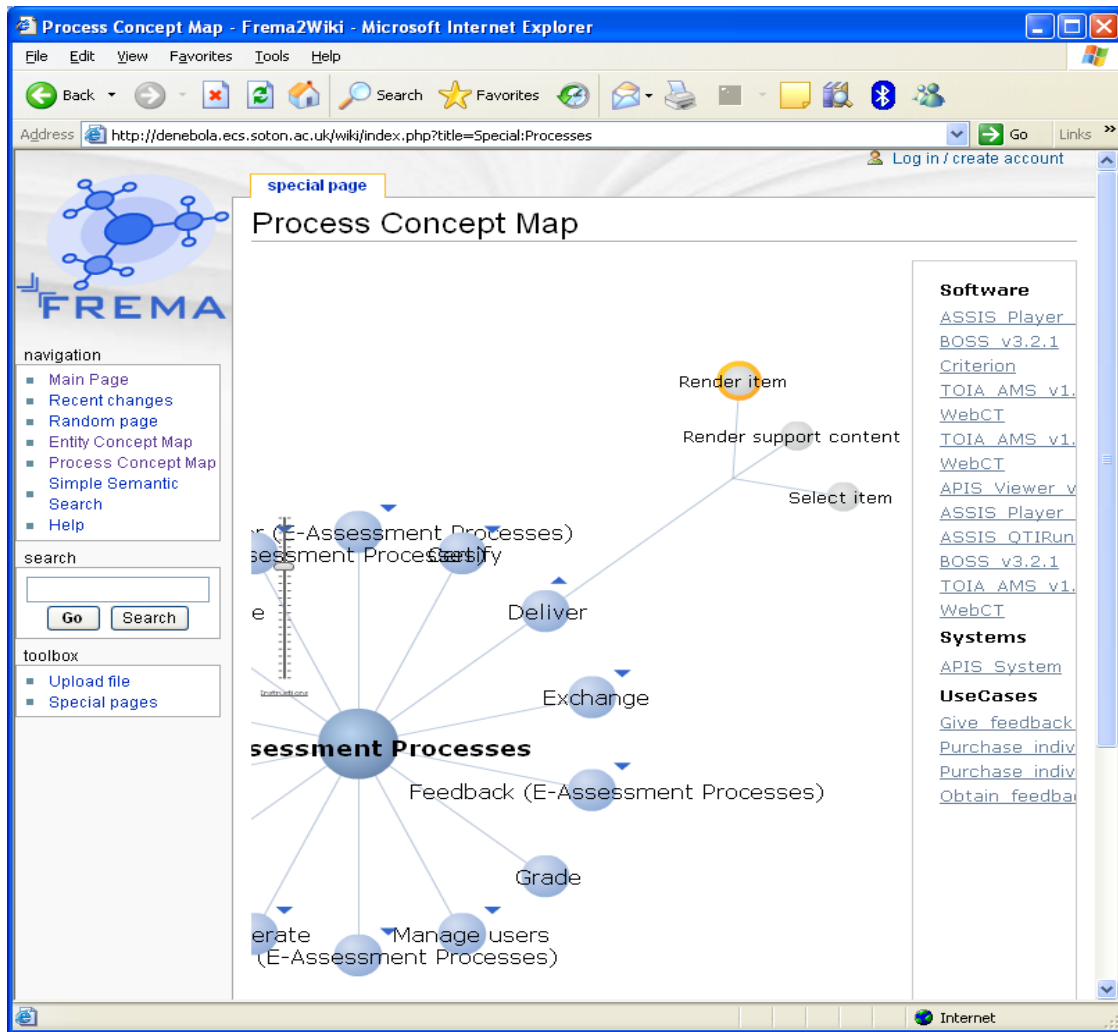


Figure 5: FREMA Semantic Wiki Concept Map Tool

We also had to extend the Wiki to support our users, who are not specialist knowledge engineers, in particular by implementing introspection of the semantic graph in order to suggest relation and class types on the editing pages. To support authors we had to be creative with the Wiki functionality, this including writing complex Wiki queries to replicate tools such as the Gap Analysis, and also creating templates for each class that automatically created attributes and common relations, including relations that tie each resource to the concept graph. We also placed creation templates (macros) on each of the class pages to help new users to create new pages by simply clicking a 'Create New X' button.

Figure 4 shows a list of resources in the FREMA SMW, Figure 5 shows the concept maps displaying information exported from the Wiki, and Figure 6 shows a resource page. The screenshots correspond to the original views shown in Figures 1,2 and 3.



Figure 6: FREMA Semantic Wiki Resource Page

6 REFLECTIONS

Our experiences of converting a Knowledge Base into a Semantic Wiki has been mostly positive, it has allowed us to open up not only the descriptions of resources, but also the whole structure of the KB. If the structural decisions about how to model the Assessment domain taken by the FREMA team turn out to be inappropriate in practice, then the community has the power to change or extend it.

However, we have had a number of difficulties. Some of these have been because of limitations in SMW, and many of these we have been able to overcome by using extensions. But some are due to the inherent differences between the Wiki approach and the way in which the Semantic Web (and KBs) typically function.

One issue has been dealing with the URI conventions used in Wikis, in particular the confusion of names and IDs. The Wiki mechanism to deal with this, disambiguation pages, works well with human

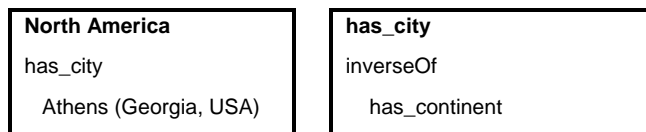
authors who are navigating the Wiki, as they can make an informed judgment (if they follow a link to a disambiguation page) as to which of the versions was relevant to the original page they were reading. However the same is not true of systems navigating the semantic graph, as the disambiguation page effectively breaks the connection. This means that the logical model is also broken, and the relation cannot be used in complex queries and inference. For example, consider the following disambiguated pages:



A query as to the location of the Parthenon will return Athens, the user will be able to follow this link to a disambiguation page, and may be able to reason about which Athens they want. However if they made the query because they did not know which Athens is relevant, then the disambiguation page cannot help them. Similarly if a machine made the more complex query to show the continent in which the Parthenon was located then the disambiguation page will break the inference chain and cause the query to return no result.

This can be fixed by updating the statement about the location of the Parthenon to refer to Athens (Greece), but disambiguation is a natural part of the evolution of Wiki content, and such updates happen in an ad-hoc manner that are not typically enforced. One solution would be to modify the reasoning engine to understand the particular meaning of *has_disambiguation* so that it can return an error message, rather than no results. This would flag to users that a query was failing, and encourage them to fix the problem.

Similarly modifying the query engine to understand the meaning of *inverseOf* would greatly increase the power of the Semantic Wiki, and reduce the problem of maintaining pairs of relations. For example, consider the following additional pages:



Now if we write a query asking for all the cities in North America (using *has_city*) we would get ‘Athens (Georgia, USA)’. However if we asked for all the cities in Europe using the same method we would not get ‘Athens (Greece)’ as a result, we would get this information only if we queried using the *has_continent* relation. We could query using both relations and then normalize the results, but building the *inverse_of* relation into the query system would solve this problem without clumsy doubled queries.

Wiki systems also resurrect arguments about the confusion of links, presentation and content, previously addressed by Open Hypermedia Systems [18] and XLink⁵. The problem with Wikis is that the simple one-mode authoring of the Wiki relies on everything being expressed in one way, and it is this simplicity that Semantic Wikis are leveraging to allow ordinary web users to author semantics. One result for the FREMA conversion was that we were unable to implement transclusions, and were forced to repeat ourselves when content appeared on more than one resource page. Experience with the Web implies that this sacrifice for simplicity is worth it, and that the value of making authoring accessible is greater than the advantage of managing links, content and presentation separately.

However, just because the editing interface views all three things in one way, does not necessarily mean that the system has to model it that way. Several Semantic Wiki implementations exist that store content and links separately (such as IkeWiki, Makna and Kaukolu), combining them at run time for both viewing and editing, but this does make the user experience a little more complicated, as they can no longer conceptualize each Wiki page as a single independent document.

A final observation of our conversion process is that the FREMA Semantic Wiki greatly benefits from being bootstrapped by a KB. The presence of an ontology in the design process means that there is an initial ontology implicitly defined in the Wiki that can be extended by a community. Crucially the bootstrapping process means that introspection tools that make recommendations about link types based on existing instances can be very effective, and we would expect this to curtail the time usually taken for a community to settle on a vocabulary for their annotations.

Wikis promote a content-first authoring paradigm, where references are not required to exist before they can be used (dangling links in Open Hypermedia parlance). In a Semantic Wiki this creates a *semantics-on-demand* environment, that is easy to author in, but can also be problematic, as the system cannot suggest relation types if the destination resource type is not known, and this reduces the ability of the system to encourage a consistent vocabulary that will eventually result in an ontology.

Using namespaces to identify page types would help solve this problem, and would also remove the issue with disambiguation breaking the logical model, but it adds complexity to the process of authoring semantics and discourages semantics-on-demand. It also embeds a class hierarchy in the naming system that may be difficult to change at a later point.

The issues with namespaces reveal that there is a tension in Semantic Wikis between the Wiki idiom and the more precise way of authoring semantics found in the Semantic Web. The same tension can be found in the merging of concepts and content, if these are separate it is more semantically accurate, but it is also harder for users to understand and to author. It seems that if you want to maintain the Wiki ease-of-use then you are forced to sacrifice some of the functionality that might otherwise help to create well-formed and useful semantic graphs.

7 CONCLUSIONS AND FUTURE WORK

In this paper we have focused on our direct experience as developers, converting from a highly managed KB to the freeform structure of a Wiki. We aimed to answer two questions:

1. What are the difficulties in *converting* between the two approaches?

⁵ XLink W3C definition: <http://www.w3.org/TR/xlink/>

2. What are the compromises made when *running* a Semantic Wiki and is the extra flexibility worthwhile?

Moving from a KB-driven web site to a Semantic Wiki means a crucial change in thinking; one must release control of the structuring ontology, and place one's faith in the wisdom of the user community. It is however a liberating experience and the potential advantages are many: a familiar Wiki editing paradigm, co-ownership of content, and evolution rather than stagnation of structures and terms.

We have demonstrated our FREMA SMW at a number of community events within the UK, and have received an overwhelmingly positive reaction, with interest partly driven by enthusiasm for the work of the FREMA project, and partly driven by interest in the Semantic Wiki itself. The loss of certain tools from the original website are more than compensated by the additional Wiki functionality, and the other advantages gained from using the mature underlying MediaWiki software, such as user and file management, administration tools, help system, discussion mechanisms and flexible configuration.

Longer term evaluation will become possible as the number of users increases. We are currently launching the public Semantic Wiki version of FREMA, and hope to undertake some evaluation of the ease of authoring and the perception of the semantics in the next few months.

We believe that for our project a Semantic Wiki can, and has, successfully replaced a knowledge-base, however we have had to sacrifice some of the semantics (in the form of disambiguation pages) and some of the tools (because of the lack of functions concerned with statistical analysis of the semantic network within the Wiki).

Analysing the Semantic Wiki implementations that are currently available has also allowed us to characterise the key features of Semantic Wikis: pages as resources, first class types, in-line annotations and typed links, and we have also been able to highlight the challenges of working with an environment that deals with semantics-on-demand.

Semantic Wikis offer a powerful new mechanism for creating semantic information in a simple way. Just as the original WikiWikiWeb made hypertext authoring a possibility for non-specialist users, so Semantic Wikis could make the Semantic Web accessible to the same group. By using evolving ontologies, and meaningful markup, users can construct semantic graphs and develop new ontologies without ever seeing an RDF statement or OWL declaration. However, there is a cost to this liberation, in that Knowledge Engineers must release control of the ontologies that guide annotation, and trust their user communities to create their own structures and evolve their own vocabularies.

8 ACKNOWLEDGMENTS

This work has been funded under the JISC project FREMA. Our thanks go to the extended project members from the Universities of Hull and Strathclyde, and to the members of the UK Assessment Community, and in particular the CETIS Assessment SIG for their help with early evaluations of our approach. We would also like to thank the NRHM reviewers for their insightful and helpful comments.

9 REFERENCES

- [1] Aumueller D., and Auer S. (2005), Towards a Semantic Wiki Experience – Desktop Integration and Interactivity in WikSAR, in Proc. of 1st Workshop on The Semantic Desktop, Galway, Ireland, Nov. 2005
- [2] Buffa, M., and Gardon, F. SweetWiki: Semantic Web Enabled Technologies in Wiki. Publisher. Mainline Group, I3S Laboratory, University of Nice, France - Acacia Group, INRIA laboratory, France. 2006.
- [3] Berners-Lee T., Hender J., and Lassila O., (2001) The Semantic Web, Scientific American, May 2001.

- [4] Campanini, S.E, Castagna P., and Tazzoli R. (2004), Platypus Wiki: a Semantic Wiki Wiki Web, Semantic Web Applications and Perspectives (SWAP) first Italian Semantic Web Workshop, Ancona, Italy
- [5] Carr, L., Hall, W., Bechhofer, S. and Goble, C. (2001) Conceptual Linking: Ontology-based Open Hypermedia. In Proceedings of Tenth International World Wide Web Conference, Hong Kong, May 1-5, pages pp. 334-342,
- [6] El-Beltagy, S. R., Hall, W., De Roure, D., and Carr, L. (2001). Linking in context. In Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia (Århus, Denmark, August 14 - 18, 2001). ACM Press, New York, NY, 151-160.
- [7] Gruber T. R. (1993). "A translation approach to portable ontologies", Knowledge Acquisition, 5(2):199-220
- [8] Hepp M., Bachlechner D., and Siorpaes K. (2005), OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis, presented at Proceedings of 2005 International Symposium on Wikis (WikiSym 2005)
- [9] Kiesel, Malte (2006), Kaukolu - Hub of the Semantic Corporate Intranet, Workshop: From Wiki to Semantics, ESWC 2006
- [10] Klein, Bertin; Höcht, Christian, Decker, Björn. Beyond Capturing and Maintaining Software Engineering Knowledge - "Wikitology" as Shared Semantics, Workshop on Knowledge Engineering and Software Engineering, KI 2005, Koblenz. 2005
- [11] Markus Krötzsch, Sebastian Schaffert, and Denny Vrandečić: Reasoning in Semantic Wikis. In: Reasoning Web, pages 310-329, ISBN 978-3-540-74613-3, LNCS 4636 (2007),
- [12] Leuf B., Cunningham W. (2001), The Wiki way: quick collaboration on the Web, Boston: Addison-Wesley
- [13] Millard, D., Bailey, C., Davis, H., Gilbert, L., Howard, Y. and Wills, G. (2006) The e-Learning Assessment Landscape. In Proceedings of International Conference on Advanced Learning Technologies (ICALT) 2006, Kerkrade, The Netherlands.
- [14] Millard, D., Howard, Y., Bailey, C., Davis, H., Gilbert, L., Jeyes, S., Price, J., Sclater, N., Sherratt, R., Tulloch, I., Wills, G. and Young, R. (2005) Mapping the e-Learning Assessment Domain: Concept Maps for Orientation and Navigation. Proceedings of e-Learn 2005, Vancouver, Canada
- [15] Millard, D. and Ross, M. (2006) Web 2.0: Hypertext by Any Other Name?. In Proceedings of ACM Conference on Hypertext and Hypermedia 2006, Odense, Denmark.
- [16] Theodore H. Nelson (1982), Literary Machines 931, Mindful Pr, 1982
- [17] Page L., Brin S., Motwani R. and Winograd T. (1999) The PageRank Citation Ranking: Bringing Order to the Web", Stanford Digital Library working paper SIDL-WP-1999-0120
- [18] Reich, S., Wiil, U. K., Nuernberg, P. J., Davis, H. C., Groenbaek, K., Anderson, K. M., Millard, D. E. and Haake, J. M. (2000) Addressing Interoperability in Open Hypermedia: the Design of the Open Hypermedia Protocol. The New Review of Hypermedia and Multimedia 5 pp. 207-248.
- [19] Schaffert S., Bischof D., Buerger T., Gruber A., Hilzensauer W., and Schaffert S. (2006), Learning with Semantic Wikis, SemWiki 06, Budva, Montenegro, June 2006
- [20] schraefel, m. c., Shadbolt, N. R., Gibbins, N., Harris, S., and Glaser, H. (2004). CS AKTive space: representing computer science in the semantic web. In Proceedings of the 13th international Conference on World Wide Web (New York, NY, USA, May 17 - 20, 2004).
- [21] Sören Auer, Jens Lehmann: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. In Franconi et al. (eds), Proceedings of European Semantic Web Conference (ESWC'07), LNCS 4519, pp. 503–517, Springer, 2007.

- [22] Souzis, A. (2005), Building a semantic wiki, *Intelligent Systems*, IEEE, vol.20, no.5pp. 87- 91, Sept.-Oct. 2005
- [23] Völkel, M., Krötzsch, M., Vrandečić, D., Haller, H., and Studer, R. (2006). Semantic Wikipedia. In *Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006)*. WWW '06. ACM Press, New York, NY, 585-5
- [24] Weal, M. J., Hughes, G. V., Millard, D. E., and Moreau, L. 2001. Open hypermedia as a navigational interface to ontological information spaces. In *Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia (Århus, Denmark, August 14 - 18, 2001)*.