

SweetWiki : Semantic WEb Enabled Technologies in Wiki

Michel Buffa, Guillaume Ereteo,
Fabien Gandon



Introduction(1)

- Wikis are great social web sites but in some cases (mostly intranets)
 - Open structure that makes navigation difficult,
 - They do not scale well,
 - They are no standards; migration is difficult
- Wikis were invented in the 90s what if we were to reinvent them now ?

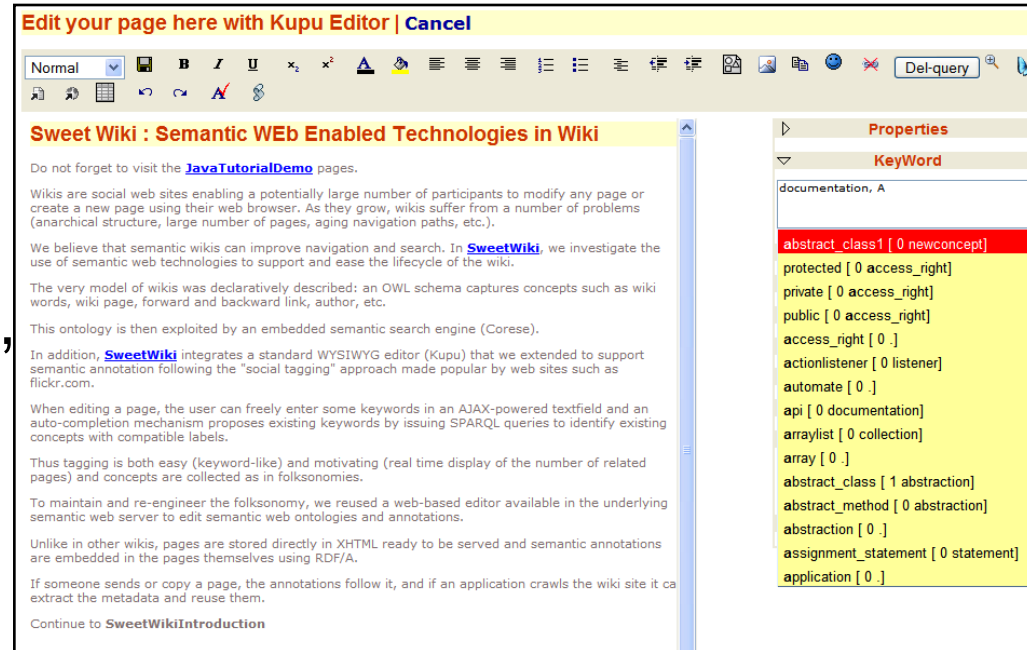
Introduction(2)

- SweetWiki is just another wiki
 - Simple to use (keep it « social »)
 - Create pages on the fly using WikiWords,
 - WYSIWYG editor for both content and metadata editing,
 - learn by « copy'n'paste from other pages »
 - AJAX enhanced wizards for linking existing local pages using WikiWords
- etc.*

The screenshot shows the SweetWiki interface. At the top, there is a navigation bar with the SweetWiki logo, a link to 'Edit this page', and the user's login status 'Logged as toto - Disconnect'. The page title is 'Sweet Wiki : Semantic WEB Enabled Technologies in Wiki'. The main content area contains several paragraphs of text, including a 'Create Page' section with a form for Name, Web, and a 'Create this page' button. The bottom of the page features a 'See Also' section with a link to 'Main.SweetWikiIntroduction', a 'Tags' information section, and a 'Page informations' section showing the author as 'toto', the web as 'Main', and the last changes as '2006-06-18'.

Introduction(3)

- Structure may be added *as needed* using **metadata**
 - Advanced support for social tagging,
 - Tag pages or *any object*
- Most metadata are implicit, **hidden** «behind the scene», used for all the internals
- May be used for displaying **dynamic content** (embedded queries)
- Keyword search (using Lucene) and **Semantic search** (using CORESE reasoning engine)



Sweet Wiki: using latest standards

- Persistence in **XHTML** (no WikiML)
- Kupu **editor** extended with **AJAX** widgets for **tagging** at editing time
- Embedded annotations in **RDFa**
- RDF/S, OWL for wiki structure and domain tags
- SPARQL **queries** for navigation, contextual template, social tagging, administration, dynamic content
- Relies on a semantic web portal framework (SeWeSe) including a reasoning engine (Corese)

All the semantic web plumbing is hidden from the user

- Don't be afraid by the Semantic Web technologies
 - We are using them because they are standards,
 - We are using very simple -easy to extend- ontologies,
- A wiki ontology (a few lines long) that describes the very model of our wiki
 - Defines what is a page, a link, a WikiWord, etc.
 - We use SPARQL queries for many standard wiki tasks
- A folksonomy built « on the fly » by the users during the social tagging approach; « cheap » ontology, but then again, we use SPARQL queries,
- In a few words : we took what we needed!

What the semantic web plumbing allows us to do...

- if you send a wiki page to someone **the annotations follow** it and can be processed by applications of the recipient;
- if an **application crawls the wiki** site it can extract the metadata and reuse them;
- if a new community of users wants a **dedicated index** just embed the corresponding SPARQL query in a wikipage;
- if the wiki engine is to be changed, the **migration transformations** can exploit the embedded metadata;
- if a users want to setup **access rules** to some documents, they can be based on these metadata merged with others;
- if some users are interested in being informed on any new information on a topic they can use **notification** systems monitoring the wiki by querying its metadata;

Corese Semantic Search Engine

RDF
annotations

RDFS/OWL
ontologies

```
select ?p display xml where
{
  ?p sw:hasForKeyWord ?k
  ?k rdfs:subClassOf jv:Widget
}
```

```
JavaGui sw:forward jv:JLabel rdfs:
JavaGui sw:author user:admin
JavaGui sw:hasForKeyWord jv:JLabel
```

```
<sew:query xslt="/xsl/tab.xsl">
select ?p display xml where
{ ?p sw:hasForKeyWord ?k
  ?k rdfs:subClassOf jv:Widget}
</sew:query>
```

Wiki pages
XHTML/JSPX +
Metadata in RDFa may
include JSP tags

```
<head xmlns:sw="http://sweetwiki.inria.fr/"
xmlns:jv=" http://www.inria.fr/acacia/java-onto#">
  <meta content="JavaGui" name="sw:name"/>
  <link href="#admin" rel="sw:author"/>
  <meta content="2006-3-2" name="sw:modification"/>
  <link href="#JavaJTable" rel="sw:forwardLink"/>
  <link href="[jv:JLabel]" rel="sw:hasForKeyWord"/>
  <link href="#JavaJPanel" rel="sw:seeAlso"/>
</head>
```


Ontology

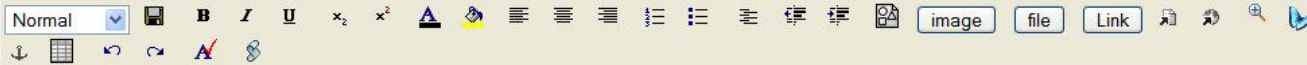
- » Add Concept
- » Merge concepts
- » Edit settings
- » **Ontology list**
- » Corese reload

Concepts Properties Settings

RDF expand all collapse all

Search Engine

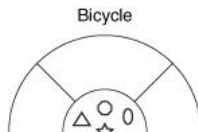
Edit your page here with Kupu Editor | Cancel



What Is Inheritance?

Generally speaking, objects are defined in terms of classes. You know a lot about an object by knowing its class. Even if you don't know what a penny-farthing is, if I told you it was a bicycle, you would know that it had two wheels, handlebars, and pedals.

Object-oriented systems take this a step further and allow classes to be defined in terms of other classes. For example, mountain bikes, road bikes, and tandems are all types of bicycles. In object-oriented terminology, mountain bikes, road bikes, and tandems are all *subclasses* of the bicycle class. Similarly, the bicycle class is the *superclass* of mountain bikes, road bikes, and tandems. This relationship is shown in the following figure.



Properties	
KeyWord	
subclass, superclass, extend, inheritance, overriding, class-hierarchy, abstract_class, j	
jbutton [1 button]	
jtree [3 gui]	
jtable [0 gui]	
jdk [0 .]	
jpanel [0 gui]	
jlabel [2 label]	
javadoc [0 documentation]	ons
java-class [0 .]	

Web server library

ed navigation, in wiki page g SPARQL kslt, etc.

as wiki pages

WYSIWYG editor for content + metadata editing with auto-completion

Social tagging/Folksonomy

- Tags may be freely entered by users,
 - AJAX autocompletion makes tagging nearly zero-cost, and unambiguous, from within the WYSIWYG editor,
 - Tag pages or any object,
 - Instant feedback at editing time of what the reader will get from your tags (motivating),
 - Advanced users may embed SPARQL queries in the pages to display dynamic results.
- Tags are implicitly organized within an ontology,
- An ontology editor with a web interface is included in the wiki; re-organizing tags impacts navigation and search

Queries in WikiPages

- Include SPARQL queries at editing time together with its XSLT rendering stylesheet
- Query embedded in XHTML as JSPX tag
- Solved (Corese) and rendered at reading time

The screenshot shows a web browser window with the title "Java Tutorial Demo Pages". The page content includes a list of links under the heading "Pages created by admin". The links are: Main, JavaMessage, JavaInheritance, SweetWikiDiscussion, MainHome, WikiWords, WikiWebs, JavaButton, SeWeSe, and JavaTutorialDemo. The page also contains some introductory text about the demo using pages from the Java Tutorial by Sun Microsystems.

Demo.

KUPU Editor - Mozilla Firefox

Eichier Edition Affichage Aller à Marque-pages Outils ?

http://localhost:8080/wiki/jsp/kupu/edit.jsp?pageToEdit=/wiki/tmp/Courses/Sar

Gmail Les serveurs de l'ESSI

Edit your page here with Kupu Editor | Cancel

Normal **B** *I* U x₂ x²

a simple bean

```
import java.awt.*;
import java.io.Serializable;

public class Simplebean extends Canvas implements Serializable
{
//Constructor sets inherited properties
public Simplebean(){
setSize(60,40);
setBackground(Color.red);
}
}
```

Social tagging

Keyword

- serializable [0 persistence]
- serialization [2 persistence]
- hashset [0 set]
- set [0 collection]
- security [0 .]
- server [0 .]

Images

Tables



nos... /wiki/data/Courses/SampleBean.jsp

this | Logged as Jeremy - [Disconnect](#)

Semantic Web Enabled Technology Wiki

Keywords

- Serialization(3)

RDFa

th... zable;

ean extends Canvas implements Serializable

herited properties

Create Page

Name

Web

setSize(60,40);
setBackground(Color.red);
}
}

See Also

Tags' informations

Category : [persistence](#),
Related tags : [Serializable](#),

RDFa

Page

- Author : [Jeremy](#)
- Web : Courses
- Last changes : 2006-03-15

Conclusion



- SweetWiki is a research project
 - Written from scratch for experimenting semantic web technologies for wikis and **other ideas**,
 - Need **to be validated** by real users; very easy to extend:
 - Add new entries in the wiki ontology (define what a « picture in a page is ») and you can add metadata, queries.
 - The reasoning engine at the heart of SweetWiki is very efficient (metadata are in memory), supports approximate searches, etc.
 - Still in alpha version, can be tried online
<http://miageprojet.unice.fr:8080>
- Other semantic wikis exist (see SemWiki 2006)

Wiki ontology

```
<rdfs:Class rdf:ID="WikiPage">
    <rdfs:subClassOf rdf:resource="#Document"/>
    <rdfs:label xml:lang="en">wiki page</rdfs:label>
    <rdfs:comment xml:lang="en">a page of a Wiki</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="camelWord">
    <rdfs:label xml:lang="en">camelword</rdfs:label>
    <rdfs:domain rdf:resource="#WikiPage"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>

<rdf:Property rdf:ID="hasForKeyWord">
    <rdfs:domain rdf:resource="#Document"/>
    <rdfs:range rdf:resource="rdfs:class"/>
    <rdfs:label xml:lang="en">hasForKeyWord</rdfs:label>
</rdf:Property>
```