Proposal and Development Status

XWiki Sakai Integration Proposal for rSmart

Contents:

1.0 Requirements: Objectives and Challenges

User Functionality and UX Needs

1. WYSIWYG Editor
   - a. Ability to toggle between the WYSIWYG editor and wiki markup without problems
   - b. Easier formatting for tables
   - c. FCKeditor, RTEditor, other WYSIWYG editors
2. Permissioning
   - a. Group aware
   - b. Support individual wiki page access restricted to one student
3. Tagging
4. Importing - import initial wiki content set up by instructor
5. Integration with other tools
   - a. Resources - ability to link to files
   - b. Gradebook - support grading of groups or individual contributions to wiki pages
   - c. Assignments - maybe allow students to link to a wiki page as a submission rather than attaching it as a word/PDF???
6. Ability to have pages with the same name without conflicts (namespaces)
7. Easier UI for permission controls
8. Easier UI for creating public views, word and PDF files
9. Ability to "control" pages and all of their "sub-pages" at one time
   - a. Convert one page and sub-pages to PDF
10. Better tracking of who changed what on a wiki page
11. Allow for public commenting without having to allow for public editing
12. Bread crumbs should not follow every step you take but just show where you are at that moment in the wiki structure.

13. Easy linking to wiki pages from other tools...similar to how you "select" files in Resources to link to text in Lessons.
14. External links on other servers
15. Archiving
16. Import/Export from/to another wiki in addition to word/pdf options
17. Searching
18. Defined UX to separate wiki use of authoring from content management of authoring
19. Copy/paste from word while maintaining styles
20. Easy creation of tables/grids/graphic organizers
21. When a wiki page is deleted it should no longer appear in the search results

2.0 XWiki Features (relating to Objectives and Challenges)

Desired Features	Feature Description
WYSIWYG Editor	- WYSIWYG, HTML, plus TWiki, Creaole 1.0, XHTML, MediWiki, Confluence, MediaWiki, JSPWiki- Full screen editing mode- Tables, embed documents and images
Permissions	- Users and Groups- Wiki, space, and page level control- View, comment, edit, delete, admin, register, program Right Management- User and Group Management console
Tagging	Unstructured and structured tagging
Namespaces	Yes
Bread crumbs	Not covered, probably not supported?
Convert pages and sub-pages to PDF	Page and include child page conversion supported, as well as, selective multi-page conversion
Versioning and rollback	- Easy view of who made changes- Compare any two versions
Comments	Add or reply to comments for: All Users, Logged Users, Guests, Admins
External links	External URLs with inline or reference file/ images
Archiving	XAR file format for archiving
Import	Office documents (word docs, spread sheets, presentations) into wiki
Export	PDF, RTF, XML, HTML, or XAR
Search	A Lucene-based plugin, supports admin and non-admin modes, selective spaces, multiple wikis, strong query syntax, indexes attachments, and provides scoring
Copy/Paste from word with styles	?
Tables/grids/graphics	Tables: sortable, filtered columns, function support, dynamic ajax loadingGraphics: SVG engine in progress

3.0 Sakai and XWiki Integration

Option 1: Have XWiki mirror Sakai's users and groups. For LDAP based Sakai deploys XWiki would use the same LDAP server to obtain group and user authentication. For supporting Sakai deploys that are using a database there would need to be synchronization between Sakai's database and XWiki's database. Logic within Sakai for adding and deleting users or groups could be used to trigger synchronization between the two applications.

For both approaches XWiki has the Rights Management component that runs within the wiki and lets users with appropriate permissions control access to a specific wiki site. Note there is no XWiki API to access administration methods. This approach would require synchronization of roles between the two applications. For LDAP deploys this would not add any overhead, but for database deploys there would need to be synchronization of roles.

Getting input from someone familiar with Sakai's database management of users and groups would help to determine the feasibility of this approach.

Option 2: After taking a closer look at a programmatic integration of Sakai and XWiki and whether to use XWiki's Java API or the RESTful API. However, using the RESTful API would be a simpler and more modular approach and would make it easier to integrate with other wikis in the future.

For either the RESTful or Java API the basic flow and breakdown of processing between Sakai and XWiki is very similar. Our proposal will address looks at the REST approach.

To handle user authentication and authorization using the REST API the request would be intercepted using the Basic HTTP Header (or alternatively a cookie) to get the user's credentials and then a call to Sakai passing that information would be made. When a user performs an operation on a wiki page the role and privileges would first make the appropriate calls to Sakai's API to return a boolean to indicate if they are authorized or not authorized to perform a specific function. In the case of a REST request the function is passed within the URL request. This approach will require synchronization of roles between Sakai and XWiki.

As mentioned in Option 1, granular permissions would be set within the XWiki administration for a given wiki site. All other functions such as tags, searches, versioning, import/exports, and the ability to manage attachments and documents would be managed by the XWiki application. The management of attachments and documents would be independent of Sakai, whether the resources reside as Sakai Resources or external to Sakai. A tighter coupling between Sakai's Resources and XWiki's documents and attachments would require additional integration work.

4.0 Summary of Development Objectives

Note that the development time estimates made below are based on our current understanding of both Sakai 2.7 and XWiki 2.3. There may be changes to subsequent versions that may have some impact on the work to be done or at a later time on work that might already done. It should also be mentioned that the XWiki community is a small community of approximately 25 members and they may not always be as responsive as we would like, in which case we may also consider spending extra time to investigate an issue here at Marist.

Original Development Summary

Phase I: (Proof of Concept)

- Deploy XWiki using browser-based RESTful API (completed)
- Integration with Sakai authentication (completed)
- Sakai Site to XWiki mapping (completed)
- Synchronize roles between Sakai and XWiki (completed)
- Testing and documentation (in progress)

Status Summary: All tasks completed with the exception of testing (85% remaining) and documentation (30% remaining).

Phase II: (Beta Release)

- Skinning XWiki, css and velocity templates (completed)
- Manage user wiki preferences within Sakai (to do)
- Unified search across Saki and XWiki (in progress)
- Unified resources across Sakai and XWiki (mostly completed)
- XWiki configuration from Sakai (in progress)
- Unified tag support (not yet addressed)
- Testing and documentation (not yet addressed)
- Sakai 2.8 integration (not yet addressed)
- Other optional features

Status Summary: 30% of task completed. Target date for completion 14 May 2011.

5.0 Phase I Development Schedule Details (Proof of Concept):

Original Phase I Development Schedule Details

Category Name	Environment Setup	
Category Id	SXW_1_1	

Requirement/Task	Deploy and setup XWiki, Sakai 2.7.1, and MySQL on two test servers
Description	
Implementation Status	Completed

Category Name	XWiki Restful API
Category Id	SXW_1_2
Requirement/Task	Work with XWiki's Restful API
Description	Login from a browser using multiple accounts with different roles, add attachments, documents, test granular permissions
Implementation Status	Completed implementation of different roles and permissions using Sakai Site roles and mapping to XWiki permissions. Page editing with RichTextEditor inside tool and ability to add any data type within the editor. This also includes local links to pages within XWiki. Will require additional testing with the addition of groups.

Category Name	Basic LTI Tool
Category Id	SXW_1_3
Requirement/Task	Make XWiki appear as a Sakai tool
Description	Using Sakai Basic LTI Portlet
Implementation Status	Integration completed as a Sakai tool, however we did not use BasicLTI, instead managed authentication directly between the Sakai Wiki tool and XWiki, transparently creating accounts in XWiki on behalf of Sakai users.

Category Name	Provisioning XWiki based on authenticated Sakai User/Roles
Category Id	SXW_1_4
Requirement/Task	XWiki Provisioning
Description	Provision authenticated Saki user requests with user, role, and site context information in payload, to provision requests on the fly in XWiki. Any other direct requests to XWiki that don't come from Sakai will be rejected. This approach would provision users with a granularity provided by Sakai. If further granularity than what is offered by Sakai today is needed, we could add these into the Sakai tool.
Implementation Status	The above has been completed. The tool transparently creates user accounts on XWiki and there is an option to either give the user access to the XWiki account credentials or not. With the credentials users can optionally log directly into XWiki, without them users can only access XWiki through Sakai.

Category Name	Phase I testing
Category Id	SXW_1_5
Requirement/Task	Test all integrated features of Phase I
Description	
Implementation Status	15% completed.

Category Name	Phase I documentation
Category Id	SXW_1_6
Requirement/Task	Technical document for Phase I implementation

Description	
Implementation Status	70% completed.

6.0 Phase II Development Schedule Details (Beta):

Original Phase II and Phase III Development Schedule Details

Category Name	XWiki skins
Category Id	SXW_2_1
Requirement/Task	Support XWiki skins and templates as other skins are stored in Sakai
Description	Add css and velocity templates to Sakai for XWiki
Implementation Status	Completed, may require some minor changes.

Category Name	Support for XWiki “All User” access
Category Id	SXW_2_2
Requirement/Task	Optionally support user access to XWiki.
Description	This would modify SXW_1_4 to allow all users of Sakai access to a wiki
Implementation Status	Completed

Category Name	A unified management of resources between Sakai and XWiki
Category Id	SXW_2_3
Requirement/Task	Synchronizing resources between XWiki and Sakai
Description	XWiki will be able to add Sakai Resources. There also needs to be the ability that when Sakai Resources are removed that they are also removed from any wikis using those resources in XWiki. Adding an external resource to XWiki would trigger an update to Sakai’s Resources.
Implementation Status	The ability to reference local links in XWiki from Sakai has been implemented. It may be a necessary to make accessible XWiki local images from Sakai (TBD). Otherwise, there really are no resources residing in XWiki that would need to be synchronized.

Category Name	XWiki user preferences
Category Id	SXW_2_4
Requirement/Task	Add XWiki user preferences to Sakai
Description	Depending on the version this may be file based or database based storage of preferences
Implementation Status	Not yet addressed. Email notifications will be added. There are Wiki and Web preferences that may be exposed. TBD

Category Name	XWiki Sakai unified search
Category Id	SXW_2_5
Requirement/Task	Unified search across Lucene indexing of Sakai and XWiki resources.
Description	Allow Sakai searches to search across Sakai and XWiki indexed resources
Implementation Status	Not yet addressed, but there doesn’t seem to be any issues with combining Sakai Resource searches with XWiki site searches.

Category Name	XWiki configuration properties
Category Id	SXW_2_6
Requirement/Task	The XWiki server IP and other properties need to use Sakai property files
Description	Have Sakai read and use properties needed for XWiki
Implementation Status	Some configuration properties have been added, but what else exactly needs to be configurable from Sakai and within XWiki still needs to be determined.

Category Name	Unified tag support across Sakai and XWiki
Category Id	SXW_2_7
Requirement/Task	Add XWiki tags as part of the unified search capability as an extension to SXW_2_5
Description	
Implementation Status	Not yet implemented.

Category Name	XWiki support for Sakai 2.8
Category Id	SXW_2_10
Requirement/Task	Ensure support/compatibility with Sakai 2.8
Description	Assume new features of 2.8 will not have a significant impact
Implementation Status	Not yet completed.

Category Name	Phase II testing
Category Id	SXW_2_11
Requirement/Task	Test all integrated features of Phase II
Description	
Implementation Status	Not completed.

Category Name	Deployment Scripts
Category Id	SXW_2_12
Requirement/Task	Create and test scripts for Sakai 2.8
Description	Create database and any other deployment related scripts
Implementation Status	Not completed.

Category Name	Phase II documentation
Category Id	SXW_2_13
Requirement/Task	Technical document for Phase II implementation
Description	
Implementation Time	Not completed.

7.0 Phase III Development Schedule Details:

Original Phase II and Phase III Development Schedule Details

8.0 Other Future Considerations

XWiki as a gadget
An interesting option, though still under discussion in the XWiki community, is exposing XWiki as a gadget. With the understanding that Sakai 3 may be considering to use Apache's Shindig container, an OpenSocial API implementation that supports the OpenSocial API and gadgets. This could have benefits when moving from Sakai 2 to Sakai 3 and greatly simplify integration of XWiki with Sakai 3. Once in place then the gadget can be embedded into an iframe. Unfortunately, at this time there has been no commitment to if and when the development enabling XWiki as a gadget will take place.

Semantic Wikis
This section is included for purposes of some forward thinking on options and considerations for Sakai3. A data-centric semantic model has some advantages to providing complex queries and information discovery. This section discusses those advantages and looks at a couple of products that provide semantic capabilities.
For semantic searches there exist a few integrated semantic wiki solutions (see Appendix A - Semantic Wikis). There also exist the option as discussed in the paper Semantic Tagging for the XWiki Platform with Zemanta and DBpedia (http://www.slideshare.net/oanat/survey-3044360) on how a wiki can use RDF/RDFS to describe component relationships and then with OWL domain vocabularies can be defined so that searches of multimedia content can be discovered and navigated.

Why Semantic Wikis
Visual Display of Information: information can be displayed by semantic maps (maps that show semantic relationships between entities), and allow for displaying of information in calendars, timelines, graphs, maps, and other views.
Automatically-generated Lists: with semantic wikis list are automatically generated; they do not need to be manually generated and prone to errors.
Improved Data Structure: the use of categories in traditional wikis for classification can become unwieldy when large and more complex categories are added. Semantic queries and the use of semantic templates and forms can avoid these complexities and offer a simple way to dynamically create new category relationships.
Improved Searches: semantic searches can implicitly leverage class and property structures to retrieve related information.
Information Reuse and Sharing: there is better sharing of information as exported data from one wiki can be imported to another wiki. Additionally, queries can be performed against other RDF/OWL wikis even if those wikis are in another language.
Extended Applications: semantic applications go beyond wiki. It can be used to better manage and query users, classes, citations, projects, papers, and other resources.

9.0 Proof of Concept (Status 2011-03-08)

Progress made on XWiki tool integration for Sakai
We have completed a proof of concept for the Sakai Wiki tool that gives Sakai users and administrators transparent access to XWiki. The functionality so far implemented is summarized in the *Tool Specifics* section below. Development work is continuing and a list of the more immediate tasks is listed in the *Tasks To Do* and *Areas of Interest* sections below. The last section, *Seeking Help In*, are some of the areas that we are looking for anyone in the community that would like to contribute in.

Tool Specifics:
- Integration with Sakai 2.7.1 and XWiki using REST API
- Integrated with Sakai Project Site to get all users and corresponding roles
- Sakai Roles are mapped to XWiki roles. Implemented as a one-to-one mapping, a given Sakai role maps to a single XWiki role. Though XWiki and Sakai have different implementations of roles, with XWiki supporting a user to have multiple roles, and Sakai, based on a site-centric model, has a single role assigned to each user for a given site. We will address this by having unique role names that include the site name in the role name, e.g. siteA.role1, siteB.role2.
- Ability to add non-site users to the wiki. These users access the wiki through XWiki.
- Ability to add wiki specific roles. These roles can be assigned to Sakai Site users or non-site users.
- The tool allows for non-site users to be added, managed, and assigned roles. These users can access the wiki directly through XWiki, instead of going through Sakai.
- A site can have one or more wikis assigned to it.
- Pages do not have to be associated with a parent page. Pages can be added to a wiki. Each page can have permissions set.
- History management presents a UI within the tool that queries and writes to the underlying XWiki application.

Tasks To Do:
- Support for other Sakai Site types in addition to Project Sites
- Have role names use a Sakai Site namespace
- Identify and add configuration properties
- We will be posting soon screenshots and a screencast of the Wiki tool

Areas Of Investigation:
- We are interested in exploring the ability to make other wiki applications have a plugin to the new wiki tool. This would implement a mapping from general wiki semantics functions to one or more calls to the underlying wiki application.

Seeking Help In:
• Help with authoring documentation that is specific to rwiki users transitioning to XWiki.
• Writing a utility to export rwiki for import into XWiki
• Authoring a Test plan
• Testing based on the Test plan
• If there is interest and we go forward with a wiki plugin tool we would like to see others use the plugin to add other wiki applications to Sakai.
• A more general question to see if there is interest within the community to use XWiki as blog tool?

Appendix A - Semantic Wikis

OntoWiki http://ontowiki.net/Projects/OntoWiki
OntoWiki facilitates the visual presentation of a knowledge base as an information map, with different views on instance data. It enables intuitive authoring of semantic content, with an inline editing mode for editing RDF content, similar to WYSIWIG for text documents. It fosters social collaboration aspects by keeping track of changes, allowing to comment and discuss every single part of a knowledge base, enabling to rate and measure the popularity of content and honoring the activity of users.
OntoWiki enhances the browsing and retrieval by offering semantically enhanced search strategies.
Other features worthy of mentioning are:
Facet-based navigation, allows users to navigate structures to select objects based on facets. Useful when users have no a priori knowledge of structures.
Map Views, bidirectional access from instance data to maps (Google maps) where latitude/longitude is available and from maps to instance details.
Calendar Views (time views), bidirectional access when date datatype exists can map from instance data to calendar views, and from calendar views to time-based facet views. Export options to iCal.
Widgets, are reusable components that users can use in a wiki. Developers can create new RDF widgets. An example of an existing widget is a conference widget that provides attribute relations to resources, such as, location, paper title, presenter, topic, etc. The concept of widgets can be extended within Sakai to for example citations where searches could return a richer, semantically qualified list of books and publications on a given subject.
Access Control, provides user, group-level access control on Models and Actions. Where Models are a given knowledge base and Actions are application specific functions or collection of functions.
Collaborative Editing, supported through the use of small data information chunks (Inline Editing) and identifying common information data (View Editing) where concepts like classes, properties, and instances links are leveraged.
SPARQL, a language for querying RDF directed, data graphs. Query examples might be, return all composer s out of Europe between 1700 and 1870. Continuing with another music example might be a query, such as Find the names of Artists that are similar to another Artist at least 80%.
Web Services for authentication (REST based), SPARQL queries, and model updates.

Semantic MediaWiki http://semantic-mediawiki.org/wiki/Semantic_MediaWiki
Semantic MediaWiki is also written in PHP. It is part of MediaWiki and is an add-on option to MediaWiki. It does not directly use RDF /OWL instead it uses a native modeling and query language, but it does allow export to RDF/OWL.
Supports categories, information that users have entered, and queries on categories and on sub-categories are inferenced. Similarly properties and types can also be queried and their inherent structure automatically leveraged (inference). Inferencing does not support the following

- Transitivity
- Inverse properties
- Domain and range restrictions
- Number restrictions and functional properties

It supports concepts which are stored or dynamic query results. Concept page can be browsed and used in other concept pages. Supports the ability to import and export ontologies.

Semantic Related Links

- /facet - a generic browser for heterogeneous semantic web repositories: http://slashfacet.semanticweb.org/
- Structured Dynamics - a suite of semantic tools: http://structureddynamics.com/products.html
- RDF (Resource Description Language) Specification: http://www.w3.org/TR/rdf-schema/
- WolframAlpha - a knowledge engine: http://www.wolframalpha.com/
- Platypus - a semantic wiki: http://platypuswiki.sourceforge.net/

Appendix B - Volunteer/Task List

Volunteer	Feature Requirements	Development	Code Review	Testing	Est. Hrs./Wk.
Alan Berg	x		x	x	
Adam Marshall	x				
David Elbert	x				
Amber Evans				x	4