Sakai Presentation Layer Redesign

Reasoning

Currently, the presentation layer of Sakai has a low level of complexity, but suffers from a design flaw that limits its use to the “single tool per page” implementation. This has caused the project developers to implement an *iframe* approach to allow navigation levels and other tools to co-exist on the same page. Not only does this cause interface design, skinning, accessibility and usability issues, it also limits the use of the tools outside of Sakai's own portal, nl. Charon.

Benefits

Redesigning the presentation layer will solve some of the current Sakai problems:

- The use of *iframes* is frowned upon by many Internet search engines and web developers. Redesigning the presentation layer will make it possible to do away with the use of *iframes*.
- Multiple, dynamic output formats can be easily developed, so that an instance of a page or tool within Sakai can easily be viewed in normal, accessibility, or printable versions.
- Layout and display can be specifically developed for mobile access, ex. a stripped down version of a site that has a very simple method of navigation, and enforces a single column of display.
- Tools can easily be externalized to other aggregation mechanisms such as an external portal without the implementation of a proxy.
- Some sections of Sakai currently embed HTML into the core Java code. With the introduction of XML into the presentation layer, these pieces of code can still exist, but produce XML rather than HTML, allowing final output to be fully transformed at the latest possible stage, paving the way for a very rich collection of skins and presentation methods.

Approach

Step one is to change tool output to not provide the caller with a full HTML page, but only a segment thereof, in clean XML output that can either partly conform to the XHTML schema, or a self developed schema that supports all the Sakai widgets and design patterns.

Step two, is to implement a compatibility layer into Charon to simply transform the output of a tool back to a single HTML page, with all the styles and Javascript requirements, allowing the output to fit back into Charon as *iframes*.

Step three, is to add two new portals into the Sakai core, replacing the existing Charon.

Production Portals

I strongly believe that there should be two production access layers within the Sakai core.

Charon Replacement

Full Sakai OOTB, having all the functionality that the current Charon provides. This portal will then take the tool output and aggregate it with any other information that should be displayed on the same page (navigation and other tools), and produce a further XML output, transforming it with the chosen XSLT skin (whether it be the normal, well known layout, or a print layout, accessibility layout etc.) on the fly, sending it to the requester (browser, PDA, cellphone, etc.) as HTML, PDF, XML, or any other output that a XSLT transformer can produce (Apache Cocoon being a good example for a stable XSLT transformation engine). The portal will be responsible for the creation of the display structure, providing the user with the ability to create sites, pages, and adding existing tools to columns on these pages. It is the final endpoint for communication between the user and the tools. The portal is also responsible for any security restrictions, and therefore provides the user with full authentication capabilities.

Remote Tool Service

An "on the fly" tool initiator, that receives a request to instantiate a tool, included in the request a unique identifier that specifies the instance to create. The request then sets up the contextual requirements for the tool on the fly, and then dispatches to the tool to retrieve its output. The output is then sent to the caller without any modification, allowing the caller to transform and aggregate the tool into a further framework. The caller would typically be a portal. Under this scenario, one would also have to implement providers in Sakai that connect to the caller for information regarding authentication or access requirements.