Research Guides and Sakaibrary

Introduction

The Sakaibrary project has an explicit goal to free library resources from silos within the library and make them available in other contexts – in this case, within the Sakai course management system. Instead of forcing users to interrupt their normal workflow to deal with dedicated library systems, selected sets of information and expertise unique to the library are being wrapped up in such a way that they can be used, seamlessly, within a student's normal course context.

For Sakaibrary as an application to function, it requires institution-independent mechanisms to gather and present library-specific data within Sakai. Once these services are available, it makes sense to explore how we can leverage them to extend the reach of library services even farther. By looking at the functionality inherent in the infrastructure necessary for Sakaibrary to function – collection and normalization of library data and services, OpenURL creation and verification, and collection of citations via metasearch — we can now ask how to effectively repurpose these services outside of the Sakai application environment.

Sakaibrary Research Guide

In trying to understand the opportunities available, we want to explore two concurrent issues:

• How to leverage the Sakaibrary infrastructure in new ways, as described above
• How to provide opportunities for instructor/librarian coordination and collaboration in bringing relevant library resources to students

A common library application that has useful overlap with existing Sakaibrary functionality is the Research Guide. In its starkest form, a research guide (or subject guide or course guide) is simply an annotated list of library resources available for work in a specific subject area or course context. These resource types – books and articles, journals (both print and electronic), online databases, tutorials, and local expertise (e.g., subject-specialists) – are common (in kind) across libraries, and some are already managed by the Sakaibrary application stack.

Additionally, integration with Sakaibrary provides two exciting possibilities. First is to use the existing (and emerging) institution-independent mechanisms used by the Sakaibrary infrastructure to both find (via search or listings) and reference (via OpenURL or other descriptors) library resources for inclusion in the guides. Second is to push the envelope regarding the specialization of library resources already begun in the Sakaibrary project. By allowing teams of instructors and librarians to "personalize" subject guides for a specific course, perhaps even to the point of including canned and constrained searches, we can provide a join locus of control where the faculty member and librarian can both expose their knowledge and expertise to students in an environment shared in common with those students.

Project goals

The goal, starkly stated, is to design an application to easily produce (and later customize) research guides, with an eye toward leveraging as much of the Sakaibrary infrastructure as possible, while allowing such guides to be viewable both within and without the Sakai framework.

Hopefully, by doing so we can create a flexible system that duplicates as little data as possible and has extensive support for creating specific types of content as explained here.

To oversimplify, the goals are:

• Provide a flexible authorization system to allow collaboration between instructors and librarians.
• Whenever possible, store generic information (ISSN) rather than site-specific information (link to a specific vendor's page for a journal) to facilitate long-lived links that work across semesters, courses, and in some cases institutions, and to avoid "linkrot."
• Have a single instance of each item that appears in a guide. Information that is used to construct a link/citation is static and used by all instances; data such as the title and descriptive text are guide-specific and can be overridden.
• Provide medium-specific editing tools for guide authors that take advantage of external services (from Sakaibrary and elsewhere) to make adding and updating items as painless as possible.
• Pursue a strategy of reuse, making it easy to use items in Sakaibrary, whole or part of another guide, and item descriptions entered elsewhere in the system.
• Provide output in a number of useful formats, including a well-described XML format that can be used by a sophisticated client (such as Sakaibrary) to take full control over the rendering of an SRG.

An overview of the architecture of the Sakaibrary Research Guide (SRG) system

The SRG will consist of four main parts:

1. An authoring system, for gathering information about included resources, authorization/authentication of authors, and storage of the guide structures themselves.
2. A Sakaibrary servlet, which will provide web-services access to data and resources already gathered and structured as part of the Sakaibrary project.
3. An optional service that can be used to perform simple transformations of data (e.g., turn an ISBN into a link to the local OPAC).
4. A rendering system (tightly coupled with the authoring system) which will produce SRG output in a variety of formats such as HTML, machine-optimized XML, PDF for printing, etc.
An overview of the interaction between the various parts of the SRG. Sakaibrary is leveraged to reduce the knowledge required by the author to information about the item (e.g., name of database) as opposed to information about the local identifier for an item (catalog number, database id, etc.).

It is intended that in addition to being a crucial part of the authoring process, Sakaibrary will serve as a target for a rendered citation (number 4) as well, taking advantage of other Sakaibrary infrastructure to provide specialized views of some items (e.g., a database could potentially include a Sakaibrary searchbox targeting that database).

**How Sakaibrary Research Guide will interact with Sakaibrary**

When possible, the SRG will rely on external services to turn generic, guide-neutral information about a resource into a specific, local pointer to that resource. Sakaibrary already collects and generates much of this information, in the form of database information, code needed to generate search boxes, OpenURL generators, etc. The Sakaibrary functionality can be coupled with other (optional) local services to allow the SRG to simply describe the resource (in much the same way a citation is a description of a specific article or book), and turn that description into a valid local pointer/link at the time the guide is viewed.

**Linking to a database via Sakaibrary**

Data flow during the authoring of an guide that includes a link to a database. Note how the knowledge needed to create this item is reduced to simply knowing the name of the database; Sakaibrary and the SRG system take care of turning that information into a full-fledged entry.

**Creating a rendered research guide**
Rendering an SRG (to HTML or other format) will reverse the process. Sakaibrary and any other external services will be called to fill in item metadata with the most recent edits (if any), making sure that links are active. This will then be output to whatever format is called for, usually HTML for a web page interaction.

![Diagram](image)

Full information about an item is pulled from the local datastore and Sakaibrary both to produce the desired output.

However, a well-specified XML format will also be available that includes all the various UIDs and other metadata available for each item. This will allow a "smart" renderer such as Sakaibrary to add searchboxes, canned and constrained searches, references to existing citation lists – anything available to Sakaibrary that makes sense in the research guide context.

At the other end of the spectrum, a print view would remove all interactive elements but include URLs in a visible form to facilitate the guides' usefulness on the printed page.

Open Question: How tightly do we integrate SRG with Sakai and Sakaibrary?

There are two primary avenues through which we could implement SRG.

**Loose coupling.** The first option, and most closely tied to the discussion above, is to implement SRG as a completely separate process, with Sakaibrary acting as a (very important) data source and a (very high-functionality) rendering target. In this model, the SRG authoring tool "lives" wherever it wants to, is implemented outside of the Sakai interface, and relies on web services to pull data from Sakaibrary. The coupling is very loose; Sakaibrary is just a pre-defined target and data-source.

**Pros and cons of the loose coupling.** This model describes an authoring tool that has the possibility of replacing Sakaibrary with another data source without drastically affecting the functionality of the system. Implementation of the authoring system can take place outside of the Sakai environment and the system/language/UI limitations that implies. The downsides are similar: all the Sakaibrary functionality is not available to the SRG authoring system (e.g., to use a citation it would have to first be put into Sakaibrary and then referenced, as opposed to searching "on the fly"), and the strengths of the Sakai system (localization, transparent data-store, etc.) would not be available. It also forces the institution to provide another environment for production, whereas we're assuming that Sakai is already up and running.

**Tight coupling.** The second is to make the SRG a part of Sakaibrary (or, perhaps, a special resource type with advanced connectivity to the Sakaibrary services), with direct access to all the functionality of Sakaibrary. In this case, what we're essentially building is the "next-generation" citation list, with hierarchy and annotations as built-in functionality. The rendering functionality would be transferred to Sakai, so external systems could still call into a servlet and get a rendering of a guide for use outside of Sakai.

**Pros and cons of the tight coupling.** Some aspects of a research guide don't fit well with the general model in Sakai the focuses on course-semesters (we want guides to live "forever") and the permissions model may need some tweaking to allow the sorts of collaboration we want. Offsetting this is the availability of all of Sakaibrary as first-order functionality in the SRG, and the ability to integrate the SRG data model with the existing Sakaibrary data model.

**Bill's thoughts**

I originally wrote this document with a focus on the loose-coupling model, but our Michigan conversation on Tuesday (Feb 20th) has convinced me that the tight coupling is more advantageous for a number of reasons:

- Creating the SRG as indicated gets us most of the way to a much more functional citation list document type "for free"  
- An SRG tightly coupled to Sakaibrary gives more opportunity for things like canned/constrained searches and forces us to start dealing with these data types sooner rather than later.  
- It avoids the overhead of designing and building a web-services infrastructure during the initial pass (although we'll likely want to think this through regardless).