Sakai as client and proof-of-concept authoring tool

A Sakai(brary)-hosted Research Guide Tool

A Research Guide can be thought of as consisting of three interconnected parts:

- The **authoring tool**, which is used to produce a dataset with appropriate references (textual and/or electronic) to available resources. Note that a single item might contain several references: a title and description, an openURL, a direct URL, etc.
- A **representation** of the research guide including all of the references in a machine-readable format.
- A **rendering** of the guide within a particular environment (web page, print, etc.) and for a particular purpose.

Goals for the data underlying the authoring environment have been spelled out elsewhere; succinctly, we want to provide specialized editors for different media/targets, and do whatever possible to make sure links and descriptions remain valid despite changes in the underlying resources' metadata (URL, title, the target library's electronic subscriptions, etc.).

Focus: the intermediate representation

While the intermediate representation is, in many ways, incidental to the important aspects of both authoring and rendering a guide, it is a useful place to start a discussion since it allows us to focus on the underlying data collected and displayed without getting distracted by the mechanisms required to actually collect or display that data.

I'll start by making some assertions as to what we're trying to build.

A **research guide** is a document that:

- is hierarchical in nature, with ordering important
- is organized by headings/subheadings to a "reasonable" depth
- contains both arbitrary (potentially marked-up) text and **guide items**

This brings up the question of what a **guide item** is.

Guide Items

A **guide item** is any item that denotes a specific resource for the guide user. An incomplete list might contain:

- Specific citations (books, articles, etc.)
- Journals
- Searchable databases (and, potentially, selected sets of databases)
- Generic individuals ("a librarian")
- Titled individuals ("this course's instructor", "subject specialist in biology", etc.)
- Specific individuals ("Bill Dueber")
- Locations (a library, academic building, office, etc.)
- Other guides (or, perhaps, sections of guides)
- Syndicated content in the form of RSS/Atom feeds

For each item, we'd like the interchange format to contain two sets of metadata:

1. **Metadata** A full set of applicable metadata in a standard format, always including a display "name" of some sort (which might be a whole APA/MLA citation for an article or book) and a description for use within this particular guide. We'll want something much lighter, I would think, than MODS or MARC. There are many citation formats to choose from, so we can avoid rolling our own. We might go with something as simple as a serialization of RIS/Refer, or more complete, like a cross of Dublin Core with OpenURL.
2. **Service Pointers** A set of URI/URLbase/key=value tuples that denote specific services and the unique data necessary to find a specific item. These will necessarily be highly context-specific (although not necessarily local – one can imagine a link into the Library of Congress for a book, for example). The idea is that a URLbase and the k/v pairs could be used to construct a URL that either (a) points at a representation of the item or its metadata (e.g., a link to a journal's homepage) or (b) allows retrieval of up-to-date metadata about an item (e.g., a link into a SakaiLibrary instance). In the latter case, it would be up to still more "out of band" agreements as to what k/v pairs are required and what format the data will be returned in.

The idea is that even if **no** Service Pointers are available to a rendering client, they'll still have enough information to provide useful representations of the items in question. The points toward using whatever standards we can that will be universally recognized (ISSN/ISBN) and perhaps helping the community to develop standard notation where none exists (e.g., UIDs for database products).

The most obvious Service Pointer is an OpenURL, constructed in such a way that the resolver can be added in by a rendering client. There's no reason not to have others, though – and no reason why de facto standards might not arise around them. The Service Pointers serve to preserve the desire to allow different services to be sources of data for the final rendering of a guide document. It will be up to the client application to decide if a particular service is available to it (not everyone can access our searchboxes) or even relevant (a link to a specific librarian might not make sense at another institution).

What we need to succeed
This list hasn’t really changed, but is perhaps understood differently in light of the above:

- **An easy-to-use authoring tool.** The best way for us to move forward with this is to leverage the existing and coming power of Sakaibrary to give authors the mechanisms to find and store Guide Items and then “re-mix” them into a research guide. Sakai provides an excellent platform on which to create a reference implementation of a Research Guide authoring tool, complete with the specialized editors we’ve talked about so much already.

- **A well-specified interchange format.** We shouldn’t get bogged down here – to be honest, it’ll be pretty easy to tweak (or wholesale change) on the fly, since we’ll be dealing mostly with internal representation in the authoring tool and the rendering client, and the export/import will be closely linked.

- **A well-thought out initial set of Service Pointers.** Beyond the obvious OpenURL and WorldCat definitions, we’ll want to think about how to define and use publicly-available, existing services (LoC, PubMed), how to expose and leverage Sakaibrary services (searchboxes, individual citations), and how to sufficiently describe and create possible highly-local services (pointers to subject-specialist librarians, for example). Some of these will clearly have to include “templates” into which a local URL (like an OpenURL resolver or a proxy server) can be put, notes on whether the data are extremely local or not (a reference to a specific citation in Sakaibrary is local to that instance, even if you’re running Sakaibrary elsewhere, but a Sakaibrary-produce Google Scholar searchbox will be equally valid from any Sakaibrary instance and running anywhere), and some redundant information (unless we want every client to have to implement an OpenURL parser to suck out the necessary data, which maybe we do).

- **A reference implementation of a client renderer, producing HTML specialized for display in Sskai (or perhaps Sakaibrary itself, though I’m imagining a separate tool at this point).**

### Data sources in and for the authoring environment

For this initial pass, I’m envisioning we’ll focus mostly on using Sakaibrary as a data source, with likely forays into the use of syndicated content. This allows us to catch a wide swatch of the data types we’re concerned with (databases, journals, and anything that can be represented as one of our citations) and instantly leverage the work we’ve done up until now.

In this sense, we can think of a guide as sort of a wrapper around a citation list. Inserting an item (and its optional description) into a guide should be able to be accomplished by any normal way of getting a citation into a citation list, as well as a few others that take advantage of other Sakaibrary services or, eventually, other outside services.

An overall look at the process is shown in the image. The authoring environment gathers metadata and references to that metadata (e.g., a Sakaibrary citation’s unique ID) and stores it, for eventual output into an XML format that can be used by a rendering environment.