Customizing digital library interfaces with Greenstone

Ian H. Witten
New Zealand Digital Library Project
Department of Computer Science
University of Waikato, New Zealand
ihw@cs.waikato.ac.nz

Digital libraries are organized, focused collections of information. They are focused on a particular topic or theme—and good digital libraries will articulate the principles governing what is included. They are organized to make information accessible in particular, well-defined, ways—and good ones will include a description of how the information is organized (Lesk, 1997).

The Greenstone digital library software is intended to help users construct simple collections of information very quickly. Indeed, only a few minutes of the user's time are needed to set up a collection based on a standard design and initiate the building process. Collections may be large—some comprise Gbytes of text; millions of documents. Furthermore, even larger volumes of information may be associated with a collection—typically audio, image, and video, with textual metadata. Once initiated, the mechanical process of building the collection may take from a few moments for a tiny collection to several hours for a multi-Gbyte one—perhaps even a day if it involves many different full-text indexes.

Naturally, collections that have idiosyncratic requirements—as most large collections do—take longer to set up, and the design and debugging process can take several days, weeks if iterative usability testing is involved. The Greenstone designers wholeheartedly endorse Alan Kay's maxim that "simple things should be simple, complex things should be possible" (Davidson, 1993).

The facilities provided, and the user interface through which they are accessed, are highly customizable at many different levels. Even impatient users—ones who set up new collections in just a few minutes—can dictate what document formats will be included (e.g. HTML, Word, PDF, PostScript, PowerPoint, Excel), what searchable indexes will be provided (e.g. full text, perhaps differentiated by language, and certain metadata such as titles), and what browsing structures will be available (e.g. list of authors, titles, classification hierarchy). More advanced, and patient, ones can control the presentation of items on the screen, personalizing any or every page that Greenstone serves up. They can translate the interface into different natural languages. If they know HTML they can hook into Greenstone widgets like the full-text search mechanism or browsers from their own pages. If they know JavaScript they can incorporate browsing mechanisms such as image maps, and using Perl they can add entirely new browsing facilities, such as stroke-based or Pinyin-based browsing for Chinese. Some new requirements are best met by altering the Greenstone "receptionist" program, written in C++, to add new facilities at runtime.

The role of metadata

A digital library's organization is reflected in the interface that it presents to users. Much of the organization rests on metadata—structured information about the information resources the library contains. Metadata is the stuff in the traditional card catalogs of bricks-and-mortar libraries (whether computerized or not). It is "structured" in that it can be meaningfully manipulated without understanding its content. For example, given a collection of source documents, bibliographic information about each document would be metadata for the collection. The structure is made plain, in terms of which pieces of text represent author names, which represent titles, and so on. The notion of metadata is not absolute but relative: it is only really meaningful in a context that makes clear what the data itself is (Lagoze and Payette, 2000). For example, given a collection of bibliographic information, metadata might comprise information about each bibliographic item, such as who compiled it and when.

The use of metadata as the raw material of organization is really the defining characteristic of digital libraries: it is what distinguishes them from other collections of online information. It is metadata that allows new material to be sited within a library and hooked into existing structures in such a way that it immediately enjoys first-class status as a member of the library. Adding new material to ordinary online information collections requires manually linking it in with existing material, but the only manual work needed when adding new items to a digital library is to determine metadata values for each one. If a standard metadata scheme is used, even that may be unnecessary: the information may already be available from another source.

Customization in Greenstone

With Greenstone, users can design their collections individually—typically by taking an existing collection that closely matches their needs and adapting its structure as necessary. The resulting design is recorded in a short file called the "collection configuration file." It specifies such things as the collection's title, its creator's email address, a description of the purpose and principles governing what is included, what input file types should be included in the collection, where the metadata comes from and what form it takes, and how the collection will look to the user. Most of the customization that non-programming users perform in Greenstone takes place in this file. It depends crucially on the availability of metadata, and the structures defined are only produced if appropriate metadata is provided.

Searching

Searching the full text of all documents in the collection is a basic facility, included by default in all collections. Collection designers can determine whether searching should be on a paragraph, section, or whole-document level (this affects the scope of matches to a given query). They can also ask for full-text indexes to be built on metadata items (e.g. titles, authors). They can split the collection into sub-collections and allow each to be searched individually, or use an automatic language identification facility to restrict searches by language.

Browsing

Greenstone includes predefined browsing structures based on certain kinds of metadata. For example, any textual metadata can be presented as an alphabetically sorted list. The list can be tabbed into alphabetic ranges, which are chosen automatically to include a reasonable number of documents in each range. The ranges are presented horizontally at the top of the screen. Date metadata can be presented in a list that allows selection by year (horizontally) and month (vertically). Metadata that has a hierarchical structure, such as library classifications, can be presented as a tree whose nodes open to reveal the data beneath. In this case the user must provide an auxiliary file giving labels for intermediate nodes of the hierarchy (e.g. subject headings corresponding to each classification number). Underlying these structures is an internal scheme of "horizontal" and "vertical" lists which are combined appropriately by the browsing mechanism.

Hierarchical phrase browsing

A novel kind of browsing is through an interactive interface to a phrase hierarchy that has been extracted automatically from the full text of a document collection (Gutwin *et al.*, 1999). It is designed to resemble a paper-based subject index or thesaurus. The user enters an initial word into a search box, and a list of phrases containing this word is shown. These phrases are minimal ones: each can be further expanded (by clicking it) into a list of phrases that contain the original one. This allows hierarchical access to the lexical content of a document collection. Ultimately you reach a leaf of the hierarchy, which takes you straight to the unique document containing that phrase.

Although designed for use with the full document text, hierarchical phrase browsing is useful on certain kinds of textual metadata, such as titles or key phrases.

Format statements

In Greenstone, format statements control the presentation to the user of each "screen" that the system generates. Format statements can be used to determine how target documents are displayed—whether they are preceded by title, for example, or indented. They control the search results page, where they determine (for example) what metadata is presented as a "snippet" that stands for matching documents, whether it should be preceded by an appropriate document icon, whether it should be a hyperlink, and what is the target of that link. In collections that provide different versions of a document (e.g. Word and the HTML that has been extracted from it), icons for both versions are often presented in the search results list so that users can choose which one they see. This is accomplished using a format statement. Format statements also apply to the browsing mechanisms mentioned above. They can be used to control how both "horizontal" and "vertical" nodes are laid out. Thus one could embolden the A-Z tabs in an alphabetically split list, or apply different formatting to the vertical and horizontal lists in a hierarchical structure.

Format statements are basically HTML, but with some additional facilities. For example, metadata values (or even the full document text) can be interpolated into a format statement. There is a conditional mechanism you can use to alter the format

depending on the contextual situation. This allows you to change the presentation of any Greenstone screen depending on the metadata values of returned documents. For example, in the search results some collections show a list of enclosing section headings within which a "hit" is nested, and the format may be different if the hit occurs at the top level of a document.

The format language is rather arcane (in future versions we plan to use XSLT instead, but it post-dates Greenstone's original design). However, the web pages that the system presents are not pre-stored but generated on the fly as needed, and the format mechanism operates at run-time. This makes it relatively easy to debug format statements: changes take effect immediately and the result can be viewed instantly.

Macros

Greenstone is a multilingual digital library system: currently there are interfaces in Arabic, Chinese, Dutch, French, German, Hebrew, Indonesian, Italian, Maori, Portuguese, Russian, Spanish, and English. To accommodate these variants, and to allow the language interfaces to be updated when new facilities are added, all web pages are passed through a macro expansion phrase before being displayed. This means that a new language can be added by providing a new set of language-specific macros, a task that has been performed many times by people with no expertise in Greenstone.

Dynamic macros

The macro facility is an extension of HTML that includes the ability to define macros and perform textual substitution. The way the digital library functionality is hooked into the user interface is through "dynamic macros" whose expansions are determined by the system (in terms of other macros). For example, the search widget is generated by a dynamic macro. Thus users can incorporate this widget into web pages of their own design, provided they go through the macro expansion phase. A total of about twenty dynamic macros provides access to Greenstone's full user interface functionality.

It is the macro language that is employed in format statements, and the metadata-substitution and conditional mechanisms mentioned above are actually embedded into this language.

Customizing Greenstone's "building" phase

Open source software permits the ultimate in customizability: changing the source code. Then, anything is possible! Greenstone operates in two phases: collection building, which is performed offline and creates the data structures necessary to support searching and browsing, and the online business of serving the collection to users. The collection building phase is written in Perl, and some customization is achieved by making modifications to this code.

All document and metadata formats are processed by Perl modules called "plug-ins", and all browsing structures are created by modules called "classifiers". Altering

existing plug-ins and classifiers, often in very small ways, provides a useful degree of customization. For example, a plug-in can define a new metadata type, which is interpreted at display time by a format statement. This enables communication between the original source document environment and what is served to users. For example, if a document is provided in different forms, these can be shown in the search results list by defining new metadata elements that contain appropriate URLs.

Similar small modifications can be made to classifiers. For example, one collection comprises mainly books but also has a few issues of different magazines. Rather than appearing in a title browser under the individual magazine name, the collection designer wanted all magazines to appear under a separate tab at the end of the A-Z array called "Magazines". When clicked a list of magazine names appears, and clicking one of these leads to a list of issues of that magazine. This was accomplished by combining the functionality of the A-Z list and hierarchical browsers, and in fact took just two extra lines of Perl in the appropriate classifier's implementation.

Altering the run-time system

The part of Greenstone that serves collections to users is called the "receptionist," and sometimes one has to resort to changing the receptionist code to achieve the desired level of customization. This rarely involves large changes, but creates software management difficulties in dealing with different parallel versions.

Our system development strategy is to accept the inevitability of occasionally having to build a special-purpose collection-dependent receptionist to achieve some desired features, and to note what is required with a view to incorporating it as an option within the standard Greenstone code.

Conclusions

There is a wide variety of different ways in which a digital library may be customized, and virtually every collection has its own idiosyncratic requirements. Although a basic Greenstone collection of new material with a standard look and feel can be set up in just a few minutes, most users require far more personalization. Of course, as the number of collections grows and the variety of styles increases, it becomes more likely that some existing collection will match new requirements.

Greenstone incorporates customization mechanisms at many levels. One of the difficulties in dealing with such a rich system is the difficulty of producing good, up to date, documentation. In fact, from a user's point of view the chief bottleneck in customization is documentation, not the facilities that are provided. Consequently collection builders need constant access to advice and assistance from others, in order to continue to learn how to tailor the software to meet ever-changing new requirements. There is a lively email discussion group for assistance with Greenstone; participants come from over 40 different countries.

Digital libraries have the advantage over other interactive systems that their user interfaces are universally based on metadata. Metadata is the glue that allows new

documents to be added and immediately become first-class citizens. It is also the key to user interface customization, and Greenstone incorporates a range of mechanisms at different levels to capitalize on this.

Acknowledgements

The Greenstone Digital Library software has grown out of the stimulating research environment of the New Zealand Digital Library project, and I gratefully acknowledge the profound influence of all project members.

References

Davidson, C. (1993) "The man who made computers personal." *New Scientist*, No. 1978, pp. 32–35; June.

Gutwin, C., Paynter, G.W., Witten, I.H., Nevill-Manning, C. and Frank, E. (1999) "Improving browsing in digital libraries with keyphrase indexes." *Decision Support Systems*, Vol. 27, No. 1/2, pp. 81–104; November.

Lagoze, C. and Payette, S. (2000) "Metadata: Principles, practices and challenges." In *Moving theory into practice: digital imaging for libraries and archives*, edited by A.R. Kenney and O.Y. Rieger. Research Libraries Group, Mountain View, CA, pp. 84–100.

Lesk, M. (1997) Practical digital libraries: Books, bytes, and bucks. Morgan Kaufmann, San Francisco.

Witten, I.H. and Bainbridge, D. (2003) *How to build a digital library*. Morgan Kaufmann, San Francisco, CA.