Greenstone: Open-Source Digital Library Software with End-User Collection Building

Ian H. Witten, David Bainbridge and Stefan J. Boddie

Department of Computer Science
University of Waikato
New Zealand
E-mail: {ihw, davidb, sjboddie}@cs.waikato.ac.nz

ABSTRACT

The Greenstone digital library software is an open-source system for the construction and presentation of information collections. Collections built with Greenstone offer effective full-text searching and metadata-based browsing facilities that are attractive and easy to use. Moreover, they are easily maintainable and can be augmented and rebuilt entirely automatically. The system is extensible: software "plugins" accommodate different document and metadata types.

Greenstone incorporates an interface that makes it easy for people to create their own library collections. Collections may be built and served locally from the user's own web server, or (given appropriate permissions) remotely on a shared digital library host. End users can easily build new collections styled after existing ones from material on the Web or from their local files (or both), and collections can be updated and new ones brought on-line at any time.

INTRODUCTION

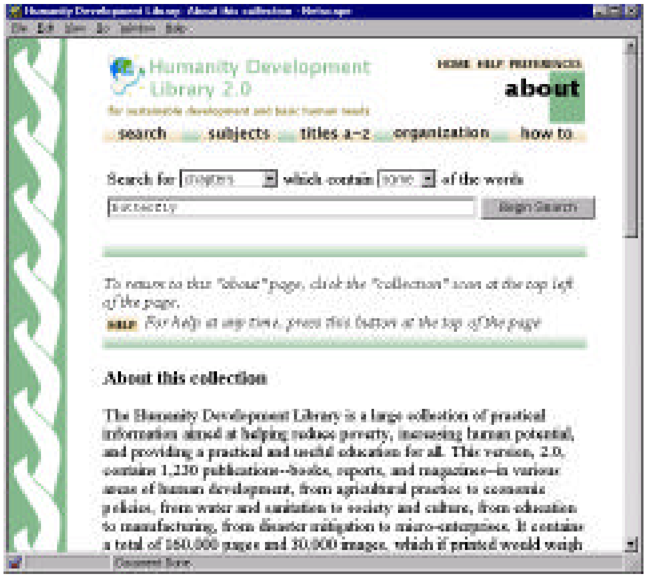
Notwithstanding intense research activity in the digital library field during the second half of the 1990s, comprehensive software systems for creating digital libraries are not widely available. In fact, the usual solution when creating a digital library is also the most obvious—just put it on the Web. But consider how much effort is involved in constructing a Web site for a digital library. To be effective it needs to be visually attractive and ergonomically easy to use, incorporate convenient and powerful searching capabilities, and offer rich and natural browsing facilities. Above all it must be easy to maintain and augment, which presents significant challenges if any manual organization is involved.

The Greenstone Digital Library Software from the New Zealand Digital Library project tackles this issue by providing a new way of organizing information and making it available over the Internet. A *collection* of information comprises several (typically several thousand, or several million) *documents et al.*, 200, and a uniform interface is provided to all documents in a collection. A library may include many different collections, each organized differently—though there is a strong family resemblance in how they are presented.

Making information available using this system is far more than "just putting it on the Web." The collection becomes searchable, browsable, and maintainable. Each collection, prior to presentation, undergoes a "building" process that, once established, is completely automatic. This process creates all the structures that are used at run-time for accessing the collection. Searching is based on various indexes, while browsing is based on various metadata such as title and author; support structures for both are created during the building operation. When new material appears it can be fully incorporated into the collection by rebuilding.

To address the exceptionally broad demands of digital libraries, the system is public and extensible. It is issued under the Gnu public license and, in the spirit of open-source software, users are invited to contribute modifications and enhancements. The system is widely used internationally—the first few responses to a call on the Greenstone mailing list for testers for a new version of the software came from India, Pakistan, US, Australia, Indonesia, South Africa, US (again), Netherlands, and Canada. Greenstone collections range from newspaper articles to technical documents, from educational journals to oral history, from visual art to folksongs. The software has been used for collections in many different languages, and for CD-ROMs that have been published by the United Nations and other humanitarian agencies in Belgium, France, Japan, and the US for distribution in developing countries.

Greenstone includes a facility for "end-user collection building". It was motivated by our work on digital libraries in developing countries, and in particular by the observation that effective human development blossoms from empowerment rather than gifting. Disseminating information originating in the developed world is very useful for developing countries. But a more effective strategy for sustained long-term development is to


Humanity Development Library: About this collection - Netscape
File Edit View Go Window Help
Humanity Development Library 2.0
for sustainable development and basic human needs
HOME HELP PREFERENCES
about
search
subjects
titles a-z
organization
how to
Begin Search
HELP
Document: Done

Search for chapters which contain some of the words

butterfly

To return to this "about" page, click the "collection" icon at the top left of the page.

For help at any time, press this button at the top of the page

About this collection

The Humanity Development Library is a large collection of practical information aimed at helping reduce poverty, increasing human potential, and providing a practical and useful education for all. This version, 2.0, contains 1,230 publications--books, reports, and magazines--in various areas of human development, from agricultural practice to economic policies, from water and sanitation to society and culture, from education to manufacturing, from disaster mitigation to micro-enterprises. It contains a total of 160,000 pages and 30,000 images, which if printed would weigh

Figure 1: Searching the HDL collection

disseminate the capability to create information collections rather than the collections themselves. This allows developing countries to participate actively in today's information society, rather than observing it from outside.

We begin by describing the facilities offered by Greenstone and showing how end users find information in collections. Next we examine the interactive interface for collection building, which extends Greenstone's domain of application by encouraging end users to build their own digital library collections. The structure of a collection is determined by its "configuration file," and we briefly show what this looks like. Finally we summarize the features of the Greenstone software.

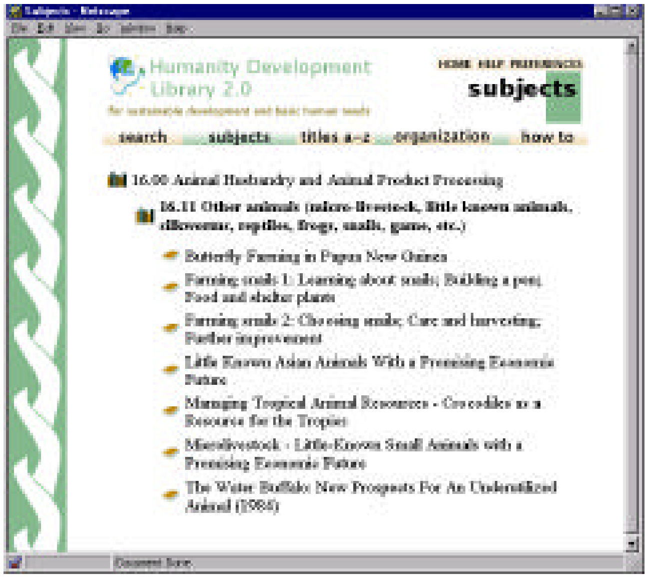
OVERVIEW OF GREENSTONE

Greenstone began in 1995 with a small group of people who wanted to make on-line technical reports more accessible to the research community by presenting them over the Web in a uniform, and fully-searchable, way. Combining skills from several areas, and using existing public-domain compression and indexing software (Witten *et al.*, 1999), a tool was devised that compiled an index from a the full text of large set of computer science technical reports gathered from many international FTP sites. Users could search for documents using any combination of words, and receive an ordered list of documents whose full text included those words, along with hyperlinks back to the original documents. The result was striking: it frequently drew attention to many extremely pertinent but previously unknown documents (such as obscure PhD theses), without the need to invest any effort in manual metadata production.

However, the tool was limited—particularly in its ability to make use of metadata that might be available—and many further issues needed to be tackled to provide a wide-ranging, general solution. Building on top of indexing and compression expertise, the New Zealand Digital Library research group is now considerably larger, mustering researchers from a wide range of areas: computer graphics, computer supported collaborative work, human computer interaction, image processing, library sciences, multimedia, ethnography, machine learning, musicology, and software engineering. The project provides a framework for a wide variety of research, and tangible end results are incorporated into the Greenstone software, which is publicly released.

Information collections built by Greenstone combine extensive full-text search facilities with browsing indexes based on different metadata types. There are several ways for users to find information, although they differ between collections depending on the metadata available and the collection design. Figures 1–4 show various stages of use; we elaborate on these below. Typically you can *search for particular words* that appear in the text, or within a section of a document, or within a title or section heading—or indeed any other metadata type. You can *browse documents by title*: just click on the displayed book icon to read it. You can *browse documents by subject*. Subjects are represented by bookshelves: click on a shelf to see the books. Where appropriate, documents come complete with a table of contents (constructed automatically): you can click on a chapter or subsection to open it, expand the full table of contents, or expand the full document.

An example of searching is shown in Figure 1 where documents in the Global Help Project's Humanity Development Library (HDL) are being searched for chapters matching the word *butterfly*. In Figure 2 the same collection is being browsed by subject: by clicking on the bookshelf icons the user has discovered an item under Section 16, Animal Husbandry. Pursuing an interest in butterfly farming, the user selects a book by clicking on


Subjects - Netscape
File Edit View Go Window Help
Humanity Development Library 2.0
for sustainable development and basic human needs
HOME HELP PREFERENCES
subjects
search
subjects
titles a-z
organization
how to
Document: Done

16.00 Animal Husbandry and Animal Product Processing

16.11 Other animals (micro-livestock, little known animals, silkworms, reptiles, frogs, snails, game, etc.)

- Butterfly Farming in Papua New Guinea
- Farming snails 1: Learning about snails; Building a pen; Food and shelter plants
- Farming snails 2: Choosing snails; Care and harvesting; Further improvement
- Little Known Asian Animals With a Promising Economic Future
- Managing Tropical Animal Resources - Crocodiles as a Resource for the Tropics
- Microlivestock - Little-Known Small Animals with a Promising Economic Future
- The Water Buffalo: New Prospects For An Underutilized Animal (1984)

Figure 2: Browsing the HDL collection by subject

its book icon. In Figure 3 the front cover of the book is displayed as a graphic on the left, and the automatically constructed table of contents appears at the start of the document. The current focus, *Introduction and Summary*, is shown in bold in the table of contents with its text starting further down the page.

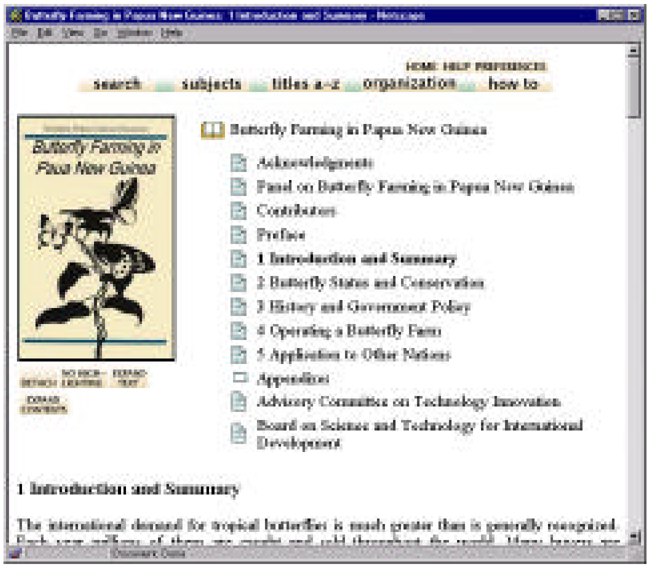
A distinction is made between *searching* and *browsing*. Searching is full-text, and—depending on the collection's design—the user can choose between indexes built from different parts of the documents, or from different metadata. Some collections have an index of full documents, an index of sections, an index of paragraphs, an index of titles, and an index of section headings, each of which can be searched for particular words or phrases. Browsing involves data structures created from metadata that the user can examine: lists of authors, lists of titles, lists of dates, hierarchical classification structures, and so on. Data structures for both browsing and searching are built according to instructions in a configuration file.

Rich browsing facilities can be provided by manually linking parts of documents together and building explicit indexes and tables of contents. However, links are difficult to maintain, and often fall into disrepair when a collection expands. Greenstone takes a different tack: it facilitates *maintainability* by creating all searching and browsing structures automatically from the documents themselves. No links are inserted by hand. This means that when new documents in the same format become available, they can be added automatically. Indeed, for some collections this is done by processes that wake up regularly, scout for new material, and rebuild the indexes—all without manual intervention.

Collections comprise many documents: thousands, tens of thousands, or even millions. Each document may be hierarchically organized into *sections* (subsections, sub-subsections, and so on). Each section comprises one or more *paragraphs*. Metadata such as author, title, date, keywords, and so on, may be associated with documents, or with individual sections of documents. This is the raw material for indexes. It must either be provided explicitly for each document and section (for example, in an accompanying spreadsheet) or be derivable automatically from the source documents. Metadata is converted into the Dublin Core standard and stored with the document for internal use.

In order to accommodate different kinds of source documents, the software is organized so that "plugins" can transform new document types into a standard XML form. Plugins exist for plain text documents; HTML, Word, PostScript and PDF files; email; and common bibliographic formats. It is straightforward to write plugins for new formats, and several have been specially written for proprietary formats. A collection may have source documents in different forms: it is just a matter of specifying all the necessary plugins. In order to build browsing indexes from metadata, an analogous scheme of "classifiers" is used: classifiers create indexes of various kinds based on metadata. Source documents are brought into the Greenstone system through a process called *importing*, which uses the plugins and classifiers specified in the collection configuration file.

The international Unicode character set is used throughout, so documents—and interfaces—can be written in any language. Collections have so far been produced in English, French, Spanish, German, Maori, Chinese, Russian, and Arabic. The New Zealand Digital Library web site (*nzdl.org*) provides numerous examples. Collections can contain text, pictures, and even audio and video clips; a text-only version of the interface is also provided to accommodate visually impaired users. Compression technology is used to ensure best use of


Butterfly Farming in Papua New Guinea: 1 Introduction and Summary - Netscape
File Edit View Go Window Help
HOME HELP PREFERENCES
search
subjects
titles a-z
organization
how to
Butterfly Farming in Papua New Guinea
DETACH
NO HIGH-LIGHTING
EXPAND TEXT
EXPAND CONTENTS
Document: Done

Butterfly Farming in Papua New Guinea

- Acknowledgments
- Panel on Butterfly Farming in Papua New Guinea
- Contributors
- Preface
- **1 Introduction and Summary**
- 2 Butterfly Status and Conservation
- 3 History and Government Policy
- 4 Operating a Butterfly Farm
- 5 Application to Other Nations
- Appendixes
- Advisory Committee on Technology Innovation
- Board on Science and Technology for International Development

1 Introduction and Summary

The international demand for tropical butterflies is much greater than is generally recognized.

Figure 3: Reading a book in the HDL

storage. Most non-textual material is either linked to textual documents or accompanied by textual descriptions (such as photo captions) to allow full-text searching and browsing.

The system includes an “administrative” function whereby specified users can examine the composition of all collections, protect documents so that they can only be accessed by registered users on presentation of a password, and so on. Logs of user activity are kept that record all queries made to every Greenstone collection (though this facility can be disabled).

Although primarily designed for Internet access over the World-Wide Web, collections can be made available, in precisely the same form, on CD-ROM. In either case they are accessed through any Web browser. Greenstone CD-ROMs operate on a standalone PC under Windows 3.X, 95/98, or NT/2000, and the interaction is identical to accessing the collection on the Web—except that response is faster and more predictable. The requirement to operate on early Windows systems is one that plagues the software design, but is crucial for many users—particularly those in underdeveloped countries seeking access to humanitarian aid collections. If the PC is connected to a network (intranet or Internet), a custom-built Web server provided on each CD makes exactly the same information available to others through their standard Web browser. The use of compression ensures that the greatest possible volume of information can be packed on to a CD-ROM.

The collection-serving software operates under Unix, Windows, and Mac OS/X, and works with standard Web servers. A flexible process structure allows different collections to be served by different computers, yet be presented to the user in the same way, on the same Web page, as part of the same digital library, even as part of the same collection. Existing collections can be updated and new ones brought on-line at any time, without bringing the system down; the process responsible for the user interface will notice (through periodic polling) when new collections appear and add them to the list presented to the user.

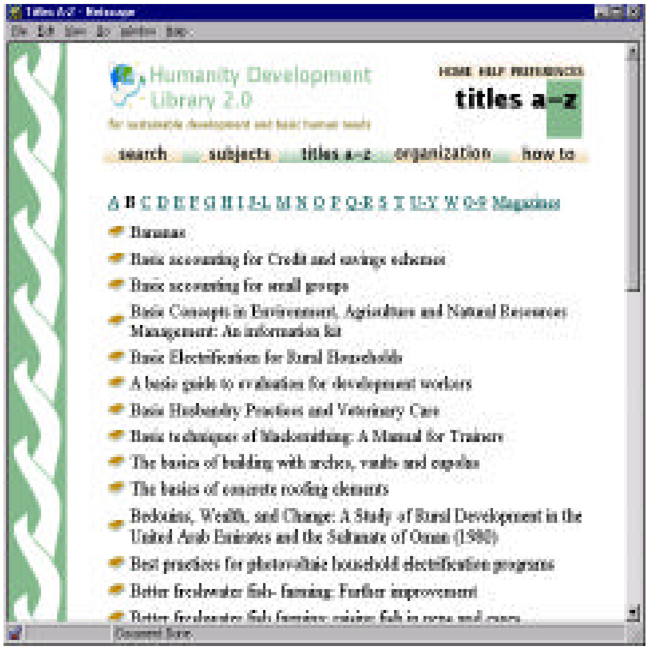
FINDING INFORMATION

Greenstone digital library systems generally include several separate collections. A home page allows you to select a collection; in addition, each collection’s “about” page (as in Figure 1) gives information about how the collection is organized and the principles governing what is included.

All icons in the screenshots of Figures 1–4 are clickable. Those at the top of the page return to the home page, provide help text, and allow you to set user interface and searching preferences. The navigation bar underneath gives access to the searching and browsing facilities, which differ from one collection to another.

Each of the five buttons provides a different way to find information. You can *search for particular words* that appear in the text from the “search” page (or from the “about” page of Figure 1). This collection contains indexes of chapters, section titles, and entire books. The default search interface is a simple one, suitable for casual users; advanced searching—which allows full Boolean expressions, phrase searching, case and stemming control—can be enabled from the *Preferences* page.

This collection has four browsable metadata indexes. You can *access publications by subject* by clicking the *subjects* button, which brings up a list of subjects, represented by bookshelves (Figure 2). You can *access publications by title* by clicking *titles a-z* (Figure 4), which brings up a list of books in alphabetic order. You can *access publications by organization* (i.e. Dublin Core “publisher”) bringing up a list of organizations. You


Titles A-Z - Netscape
File Edit View Go Window Help
Humanity Development Library 2.0
for sustainable development and basic human needs
HOME HELP PREFERENCES
titles a–z
search
subjects
titles a–z
organization
how to
A B C D E F G H I J-L M N O P Q-R S T U-Y W 0-9 Magazines
Document: Done

- Bananas
- Basic accounting for Credit and savings schemes
- Basic accounting for small groups
- Basic Concepts in Environment, Agriculture and Natural Resources Management: An information kit
- Basic Electrification for Rural Households
- A basic guide to evaluation for development workers
- Basic Husbandry Practices and Veterinary Care
- Basic techniques of blacksmithing: A Manual for Trainers
- The basics of building with arches, vaults and cupolas
- The basics of concrete roofing elements
- Bedouins, Wealth, and Change: A Study of Rural Development in the United Arab Emirates and the Sultanate of Oman (1980)
- Best practices for photovoltaic household electrification programs
- Better freshwater fish- farming: Further improvement
- Better freshwater fish farming: raising fish in pens and cages

Figure 4: Browsing titles in the HDL

access publications by "how to" listing, yielding a list of hints defined by the collection's editors. We use the Dublin Core as a base and extend it in an *ad hoc* manner to accommodate the individual requirements of collection designers.

THE COLLECTOR

In Greenstone, the structure of a particular collection is determined when the collection is set up. This includes such things as the format, or formats, of the source documents, how they should be displayed on the screen, the source of metadata, what browsing facilities should be provided, what full-text search indexes should be provided, and how the search results should be displayed. Once the collection is in place, it is easy to add new documents—so long as they have the same format as the existing documents, and the same metadata is provided, in exactly the same way.

The Greenstone "Collector" has the following functions:

- create a new collection with the same structure as an existing one;
- create a new collection with a different structure;
- add new material to an existing collection;
- modify the structure of an existing collection;
- delete a collection;
- write an existing collection to a self-contained, self-installing Windows CD-ROM.

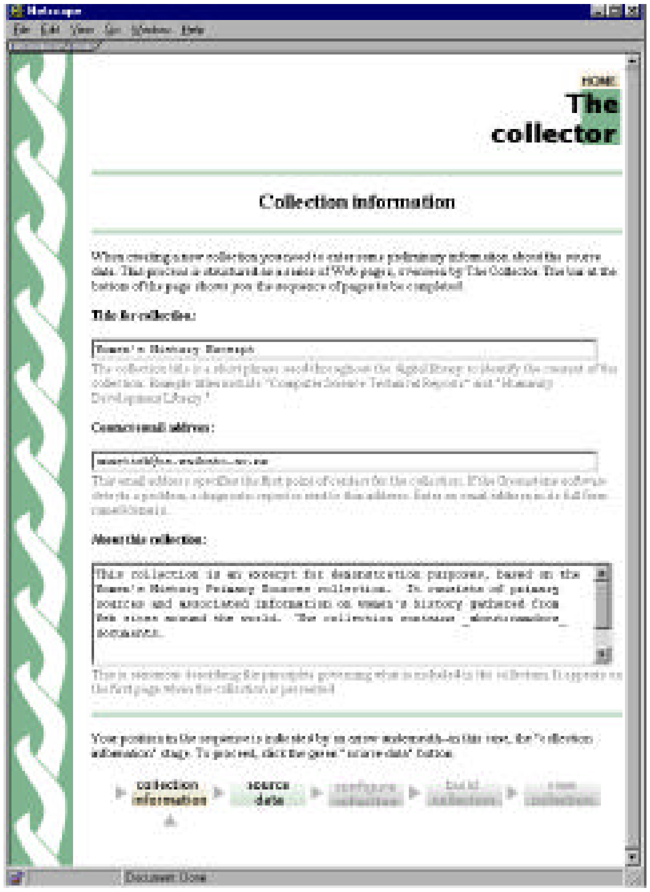
We will walk through the process of using the Collector to create a new collection, in this case from a set of HTML files stored locally. First, an explanatory Web page appears that asks the user to decide first whether to work with an existing collection or build a new one. The former case covers the first two options above; the latter covers the remainder.

Logging in

Either way it is necessary to log in before proceeding. Note that in general, users access the collection-building facility remotely, and build the collection on a Greenstone server. Of course, we cannot allow arbitrary people to build collections (for reasons of propriety if nothing else), so a built-in security system forces people who want to build collections to log in first. This allows a central system to offer a service to those wishing to build information collections and use that server to make them available to others. Alternatively, a user who is running Greenstone on his or her own computer may build collections locally, but it is still necessary to log in because other people who view these Web pages should not be allowed to build collections.

Dialog structure

Upon completion of login, a new page appears that shows the sequence of steps involved in collection building:



Collection information

When creating a new collection you need to enter some preliminary information about the source data. This process is structured as a series of Web pages, overseen by The Collector. The bar at the bottom of the page shows you the sequence of pages to be completed.

Title for collection:

Women's History Excerpt

The collection title is a short phrase used throughout the digital library to identify the content of the collection. Example titles include "Computer Science Technical Reports" and "Humanity Development Library."

Contact email address:

[illegible]

This email address specifies the first point of contact for the collection. If the Greenstone software detects a problem, a diagnostic report is sent to this address. Enter an email address in its full form: name@domain.

About this collection:

This collection is an excerpt for demonstration purposes, based on the Women's History Primary Sources collection. It consists of primary sources and associated information on women's history gathered from Web sites around the world. The collection contains [illegible] documents.

This is a statement describing the principles governing what is included in the collection. It appears on the first page when the collection is presented.

Your position in the sequence is indicated by an arrow underneath—in this case, the "collection information" stage. To proceed, click the green "source data" button.

Figure 5: A typical stage in using the Collector

1. Collection information
2. Source data
3. Configuring the collection
4. Building the collection
5. Viewing the collection.

The first step is to specify the collection's name and associated information—the screenshot in Figure 5 gives the flavor of the interaction. The second step is to say where the source data is to come from. The third is to adjust the configuration options, which requires considerable understanding of what is going on—it is really for advanced users only. The fourth step is where all the (computer's) work is done. During this "building" process the system makes all the indexes and gathers together any other information that is required to make the collection operate. The fifth step is to check out the collection that has been created.

These five steps are displayed as a linear sequence of buttons at the bottom of each page generated by the Collector. This display helps users keep track of where they are in the process. The button that should be clicked to continue the sequence is shown in green; the other buttons are grayed out because they are inactive. The buttons change to yellow as the user proceeds through the sequence, and they can return to an earlier step by clicking the corresponding yellow button. This display is modeled after the "wizards" that are widely used in commercial software to guide users through the steps involved in installing new software.

Collection information

The next step, collection information (shown in Figure 5), is to enter some information about the new collection:

- title,
- contact email address,
- brief description.

The collection title is a short phrase used throughout the digital library to identify the content of the collection: our collections include titles like *Food and Nutrition Library*, *World Environmental Library*, and so on. The email address specifies the first point of contact for any problems encountered with the collection. If the

Greenstone software detects a problem, a diagnostic report is sent to this address. Finally, the brief description is a statement that appears under the heading *About this collection* on the collection's home page (e.g. Figure 1).

The user's current position in the collection-building sequence is indicated by an arrow that appears in the display at the bottom of each screen—in this case the *collection information* stage. The user proceeds by clicking the green button, labeled *source data*.

Source data

At this point the user specifies the source text that comprises the collection. Either a new collection is created, or an existing one is "cloned." Creating a totally novel collection with a completely different structure from existing ones can be a major undertaking, and the most effective way to create a new collection is usually to base its structure on an existing one, that is, to clone it.

When cloning, the choice of current collections is displayed on a pull-down menu. Since there are usually many different collections, there is a good chance that a suitable structure exists. The document file types in the new collection should be amongst those catered for by the old one, the same metadata should be available, and the metadata should be specified in the same way; however, Greenstone is equipped with sensible defaults. For instance, if document files with an unexpected format are encountered, they will simply be omitted from the collection (with a warning message for each one). If the metadata needed for a particular browsing structure is unavailable for a particular document, that document will be omitted from the structure.

The alternative to cloning an existing collection is to create a completely new one. A bland collection configuration file is provided that accepts a wide range of different document types and generates a searchable index of the full text and an alphabetic title browser. Title metadata is available for many document types, such as HTML, email, and Microsoft WORD—note, however, that for WORD Greenstone uses the system's "Summary" information for the file, which is frequently incorrect because many users ignore this Microsoft feature.

Boxes are provided to indicate where the source documents are located: any number of input sources can be specified. There are three kinds of specification:

- a directory name on the Greenstone server system (beginning with "file://")
- an address beginning with "http://" for files to be downloaded from the Web
- an address beginning with "ftp://" for files to be downloaded using FTP.

In each case of "file://" or "ftp://" the collection will include all files in the specified directory, any directories it contains, any files and directories *they* contain, and so on. If instead of a directory a filename is specified, that file alone will be included. For "http://" the collection will mirror the specified Web site.

Configuring the collection

The construction and presentation of all collections is controlled by specifications in a configuration file (see below). Advanced users may use the next page to alter the configuration settings. Most, however, will proceed directly to the final stage, and the interface anticipates this by letting the user bypass this stage altogether.

Building the collection

The next stage is where the computer "builds" the new collection. Up until now, the responses to the dialog have merely been recorded in a temporary file. The building stage is where the action takes place.

First, an internal name is chosen for the new collection, based on the title that has been supplied (and avoiding name clashes with existing collections). Then a directory structure is created that includes the necessary files to retrieve, index and present the source documents. To retrieve source documents already on the file system, a recursive file system copy command is issued; to retrieve offsite files a web mirroring package is used to recursively copy the specified site along with any related image files.

Next, the documents are converted into a standard XML form. Appropriate plugins to perform this operation must be specified in the collection configuration file. This done, the copied files are deleted: the collection can always be rebuilt, or augmented and rebuilt, from the information stored in the XML files.

Then the full-text searching indexes, and the browsing structures, specified in the collection configuration file are created. Finally, assuming that the operation has been successful, the contents of the building process is moved to the area for active collections. This precaution ensures that if a version of this collection already exists, it continues to be served right up until the new one is ready. Use of global, persistent document identifiers ensures the changeover is almost always invisible to users.

The building stage is potentially very time-consuming. Small collections take a minute or so but large ones can take a day or more. The Web is not a supportive environment for this lengthy kind of activity. While a button is from leaving the building page, and no way to detect if they do. In this case the Collector continues building the collection regardless and installs it when building terminates.

Progress is displayed in a status area at the bottom of the building screen, updated every five seconds. Warnings are written if input files or URLs are requested that do not exist, or exist but there is no plugin that can process them, or the plugin cannot find an associated file, such as an image file embedded in a HTML document. The intention is that the user will monitor progress by keeping this window open in their browser. If any errors cause the process to terminate, they are recorded in this status area.

Viewing the collection

When the collection is built and installed, a *View collection* button becomes active. This takes the user directly to the newly built collection.

Finally, email is sent to the collection's contact email address, and to the digital library administrator, whenever a collection is created (or modified). This allows those responsible to check when changes occur, and monitor what is happening on the system.

Working with existing collections

Four further facilities are provided when working with an existing collection: add new material, modify its structure, delete it, and write it to a self-contained, self-installing CD-ROM.

To add new material to an existing collection, the same dialog structure is used, but entry is at the "source data" stage. The new data that is specified is copied as before and converted to XML, joining any existing imported material.

Revisions of old documents should perhaps replace existing ones rather than being treated as entirely new. However, this is so difficult to determine that all new documents are added to the collection unless they are textually identical to existing ones. While an imperfect process, in practice the browsing structures are sufficiently clear to make it straightforward to ignore near-duplicates. The aim of the Collector is to support the most common collection-building tasks in a straightforward manner—more careful updating is possible through a suite of command-line scripts.

To modify the structure of an existing collection essentially means to edit its configuration file. If this option is chosen, the dialog is entered at the "configuring the collection" stage.

To delete a collection, simply select it from a list and confirm its deletion. This is not as foolhardy as it might seem, for only collections that were built by the Collector can actually be removed—other collections (typically built by advanced users working from the command line) are not included in the selection list. It would be nice to be able to selectively delete material from a collection through the Collector, but this functionality does not yet exist. At present this must be done from the command line by inspecting the file system.

Finally, in order to write an existing collection to a self-contained, self-installing CD-ROM, the collection's name is specified and it is massaged into a disk image in a standard directory.

THE COLLECTION CONFIGURATION FILE

Figure 6 shows a sample collection configuration file. Some of the information in the file (*e.g.* the email address at the top, the collection name and description near the bottom) was gathered from the user during the Collector dialog. The *indexes* line builds a single index comprising the text of all the documents. The *classify* line builds an alphabetic classifier of the title metadata.

The list of plugins is designed to be reasonably permissive. For example, ZIPPlug uncompresses any Zipped files, and because plugins operate in a pipeline the output of this decompression will be available to the other plugins. GMLPlug ensures that any documents previously imported into the collection will be processed properly when the collection is rebuilt ("GML" is the name given to our internal XML document format). TEXTPlug, HTMLPlug and EMAILPlug process documents of the appropriate types, identified by their file extension. RecPlug (for "recursive") expands subdirectories and pours their contents into the pipeline, ensuring that arbitrary directory hierarchies are traversed.

```
creator         annetteb@cs.waikato.ac.nz
maintainer      annetteb@cs.waikato.ac.nz
public          true
beta            true

indexes         document:text
defaultindex    document:text

plugin          ZIPPlug
plugin          GMLPlug
plugin          TEXTPlug
plugin          HTMLPlug –file_is_url
plugin          EMAILPlug
plugin          ArcPlug
plugin          RecPlug


classify        AZList metadata=Title

collectionmeta collectionname    "Women’s History Excerpt"
collectionmeta collectionextra   "This collection is an excerpt for demonstration purposes, based on the\
                                 Women’s History Primary Sources collection.  It consists of primary\
                                 sources and associated information on women’s history gathered from\
                                 Web sites around the world.  The collection contains _about:numdocs_\
                                 documents"
collectionmeta .document:text    "documents"
```

Figure 6: Configuration file for a simple example collection

More indicative of Greenstone’s power than the generic structure in Figure 5 is the ease with which other facilities can be added. To choose just a few examples:

- A full-text-searchable index of titles could be added by augmenting the *indexes* line with one extra item.
- If authors’ names were encoded in the Web pages using the HTML metaname construct, a corresponding index of authors could also be added by expanding the *indexes* line
- With author metadata, an alphabetic author browser would require an additional *classify* line.
- WORD and/or PDF documents could be included by specifying the appropriate plugins
- Language metadata could be inferred by specifying an “extract-language” option to each plugin
- With language metadata present, a separate index could be built for document text in each language
- Acronyms could be extracted from the text automatically and a list of acronyms added
- Keyphrases could be extracted from each document and a keyphrase browser added
- A phrase hierarchy could be extracted from the full text of the documents and made available for browsing
- The format of any of these browsers, or of the documents themselves when they were displayed, or of the search results list, could all be altered by appropriate “format” statements.

Skilled users could add any of these features to the collection by making a small change to the information presented during the “Configuring the collection” stage. However, we do not anticipate that many casual users will operate at this level. More likely, someone who wants to build new collections of a certain type will arrange for an expert to construct a prototype collection with the desired structure, and proceed to clone that into further collections with the same structure but different material.

SUMMARY OF FACILITIES

We close with a brief summary of Greenstone facilities, many of which have not been explicitly mentioned above.

Accessible via Web browsers. Collections are accessed through a standard web browser. The browser is used for both local and remote access—whether Greenstone is running on your own personal computer or on a remote central library server.

Runs on Windows and Unix. Collections can be served on either Windows (3.1/3.11, 95/98, NT/2000) or Unix (Linux, SunOS, OS/X). Any of these systems serve Greenstone collections over the Internet using either an integrated built-in Web server or an external Web server.

Full-text and fielded search. Users can search the full text of the documents, or choose between indexes built from different parts of the documents. Queries can be ranked or Boolean; terms can be stemmed or unstemmed, case-folded or not.

Flexible browsing facilities. The user can browse lists of authors, lists of titles, lists of dates, hierarchical classification structures, and so on. Different collections offer different browsing facilities.

Creates access structures automatically. Searching and browsing structures are built directly from the documents themselves: no links are inserted by hand. If new documents in the same format become available, they can be merged into the collection automatically. Existing hypertext links in the original documents, leading both within and outside the collection, are preserved.

Makes use of available metadata. Metadata forms the raw material for browsing indexes: it may be associated with each document or with individual sections within documents. Metadata must be provided explicitly (often in an accompanying spreadsheet) or derivable automatically from the source documents. The Dublin Core scheme is used, with provision for extensions.

Plugins and classifiers extend the system's capabilities. Plugins can be written to accommodate new document types. Classifiers can be written to create new kinds of browsing indexes based on metadata.

Multiple-language documents. Unicode is used throughout the software, allowing any language to be processed in a consistent manner, and searched properly. On-the-fly conversion is used to convert from Unicode to an alphabet supported by the user's Web browser. A "language identification" plugin allows automatic identification of languages in multilingual collections, so that separate indexes can be built.

Multiple-language user interface. The interface can be presented in multiple languages.

Multimedia collections. Greenstone collections can contain text, pictures, audio and video clips. Most non-textual material is either linked in to textual documents or accompanied by textual descriptions (ranging from figure captions to descriptive paragraphs) to allow full-text searching and browsing. However, the architecture is general enough to permit implementation of plugins and classifiers for non-textual data.

Classifiers allow hierarchical browsing. Hierarchical phrase and keyphrase indexes of text, or indeed any metadata, can be created using standard classifiers. Such interfaces are described by Paynter *et al*. (2000).

Designed for multi-gigabyte collections. Collections can contain millions of documents, making the Greenstone system suitable for collections up to several gigabytes. Compression is used to reduce the size of the indexes and text. Small indexes have the added bonus of faster retrieval.

New collections appear dynamically. Collections can be updated and new ones brought on-line at any time, without bringing the system down; the process responsible for the user interface will notice (through periodic polling) when new collections appear and add them to the list presented to the user.

Collections can be published on CD-ROM. Greenstone collections can be published, in precisely the same form, on a self-installing CD-ROM. The interaction is identical to accessing the collection on the Web (Netscape is provided on each disk)—except that response times are faster and more predictable. For collections larger than one CD-ROM, a multi CD-ROM solution has been implemented.

Distributed collections are supported. A flexible process structure allows different collections to be served by different computers, yet be presented to the user in the same way, on the same Web page, as part of the same digital library (Bainbridge *et al.*, 2001). The Z39.50 protocol is also supported, both for accessing external servers and (under development) for presenting Greenstone collections to external clients.

What you see—you can get! The Greenstone Digital Library is open-source software, available from the New Zealand Digital Library (*nzdl.org*) under the terms of the GNU General Public License. The software includes everything described above: Web serving, CD-ROM creation, collection building, multi-lingual capability, plugins and classifiers for a variety of different source document types. It includes an autoinstall feature to allow easy installation on both Windows and Unix.

REFERENCES

Bainbridge, D., Witten, I.H., Buchanan, G., McPherson, J., Jones, S. and Mahoui, A. (2001) "Greenstone: A platform for distributed digital library applications." *Proc European Digital Library Conference*, Darmstadt, Germany; September.

Paynter, G.W., Witten, I.H., Cunningham, S.J. and Buchanan, G. (2000) "Scalable browsing for large collections: a case study." *Proc Fifth ACM Conference on Digital Libraries*, San Antonio, TX, pp. 215–223; June.

Witten, I.H., Moffat, A. and Bell, T.C. (1999) *Managing gigabytes: Compressing and indexing documents and images.* Morgan Kaufmann, San Francisco, CA.