DOCUMENT RESUME

ED 459 820 IR 058 356

AUTHOR	Witten, Ian H.; Bainbridge, David; Boddie, Stefan J.
TITLE	Power to the People: End-User Building of Digital Library Collections.
PUB DATE	2001-06-00
NOTE	12p.; In: Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (1st, Roanoke, Virginia, June 24-28, 2001). For entire proceedings, see IR 058 348. Figures may not reproduce well.
AVAILABLE FROM	Association for Computing Machinery, 1515 Broadway, New York NY 10036. Tel: 800-342-6626 (Toll Free); Tel: 212-626-0500; e-mail: acmhelp@acm.org. For full text: http://www1.acm.org/pubs/contents/proceedings/dl/379437/.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Access to Information; *Computer Interfaces; Computer Software Development; *Electronic Libraries; Information Systems; Internet; Library Collection Development; User Needs (Information); *Users (Information); World Wide Web

ABSTRACT

Digital library systems focus principally on the reader: the consumer of the material that constitutes the library. In contrast, this paper describes an interface that makes it easy for people to build 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, can collections can be updated and new ones brought online at any time. The interface, which is intended for non-professional end users, is modeled after widely used commercial software installation packages. The paper also describes an interface for the administrative user who is responsible for maintaining a digital library installation. (Author/AEF)

Power to the People: End-User Building of Digital Library Collections

By: Ian H. Witten, David Bainbridge & Stefan J. Boddie

Power to the People: End-user Building of Digital Library Collections

Ian H. Witten
Dept of Computer Science
The University of Waikato
Hamilton, New Zealand
+64 7 838 4246
ihw@cs.waikato.ac.nz

David Bainbridge
Dept of Computer Science
The University of Waikato
Hamilton, New Zealand
+64 7 838 4407
davidb@cs.waikato.ac.nz

Stefan J. Boddie
Dept of Computer Science
The University of Waikato
Hamilton, New Zealand
+64 7 838 6038
sjboddie@cs.waikato.ac.nz

ABSTRACT

Naturally, digital library systems focus principally on the reader: the consumer of the material that constitutes the library. In contrast, this paper describes an interface that makes it easy for people to build 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. The interface, which is intended for non-professional end users, is modeled after widely used commercial software installation packages. Lest one quail at the prospect of end users building their own collections on a shared system, we also describe an interface for the administrative user who is responsible for maintaining a digital library installation.

1. INTRODUCTION

The Greenstone Digital Library Software from the New Zealand Digital Library (NZDL) project provides a new way of organizing information and making it available over the Internet. A *collection* of information is typically comprised of several thousand or several million *documents*, 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.

Greenstone collections are widely used, with many of them publicly available on the Web. Some have also been written, in precisely the same form, to CD-ROMs that are widely distributed within developing countries (50,000 copies/year). Created on behalf of organizations such as UNESCO, the Pan-American Health Organization, the World Health Organization, and the United Nations University, they cover topics ranging from basic humanitarian needs through environmental concerns to disaster relief. Titles include the *Food and Nutrition Library*, *World Environmental Library*, *Humanity Development Library*, *Medical and Health Library*, *Virtual Disaster Library*, and the *Collection on Critical Global Issues*. Further details can be obtained from *www.nzdl.org*.

A recent enhancement to Greenstone is a facility for what we call "end-user collection building." It was provoked 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, as the above-mentioned collections do, is very useful for developing countries. But a more effective strategy for sustained long-term development is to disseminate the capability to create information collections rather than the collections themselves [7]. This allows developing countries to participate actively in our information society rather than observing it from outside. It will stimulate the creation of new industry. And it will help ensure that intellectual property remains where it belongs—in the hands of those who produce it.

The end-user collection building facility, which we call the "Collector," is modeled after popular end-user installation software (such as InstallShield[1]). Frequently called a software "wizard"—a term we deprecate because of its appeal to mysticism and connotations of utter inexplicability—this interaction style suits novice users because it simplifies the choices and presents them very clearly.

The core Greenstone software addresses the needs of the reader: the Collector addresses the needs of people who want to build and distribute their own collections. A third class of user, vital in any multi-user Greenstone installation is the system administrator, who is responsible for configuring the software to suit the local environment, enabling different classes of Greenstone user, setting appropriate file permissions, and so on. Greenstone includes an interface, not described in previous papers, through which the administrative user can check the status of the system, and alter it, interactively. Sensitive and flexible administrative support becomes essential when many end users are building collections.

This paper begins with a brief synopsis of the features of Greenstone. To some extent this section overlaps material presented at DL00 [8], but many features are new and others extend what was previously reported. The remainder of the paper is completely new. We examine the new interactive interface for collection building, which will extend Greenstone's domain of application by encouraging end users to create their own digital library collections. The structure of a collection is determined by

JCDL '01, June 24-28, 2001, Roanoke, Virginia, USA.

[1] www.installshield.com

its "collection configuration file," and we briefly examine what can be specified in this file. Next we turn to the administrator's interface and describe the facilities it provides. Finally we discuss the design process and usability evaluation of the system.

2. THE GREENSTONE SOFTWARE

To convey the breadth of coverage provided by Greenstone, we start with a brief overview of its facilities. More detail appears in [8].

Accessible via Web browsers. Collections are accessed through a standard web browser, such as Netscape or Internet Explorer. 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) or Unix (Linux and SunOS). Any of these systems serve Greenstone collections over the Internet using either an integrated built-in Web server (the "local library" version of Greenstone) or an external server—typically Apache (the "web library" version).

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. Some collections have an index of full documents, an index of sections, an index of paragraphs, an index of titles, and an index of authors, each of which can be searched for particular words or phrases. 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, and even within a collection a broad variety of browsing interfaces are available. Browsing and searching interfaces are constructed during the building process according to collection configuration information.

Creates access structures automatically. All collections are easy to maintain. Searching and browsing structures are built directly from the documents themselves: no links are inserted by hand. This means that if new documents in the same format become available, they can be merged into the collection automatically. However, 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, however, provision is made for extensions, and other schemes.

Plugins and classifiers extend the system's capabilities. "Plugins" (small modules of PERL code) can be written to accommodate new document types. Existing plugins process plain text documents, HTML documents, Microsoft WORD, PDF, PostScript, and some proprietary formats. So collections can include different source document types, a pipeline of plugins is formed and each document passed down it; the first plugin to recognize the document format processes it. Plugins are also used for generic tasks such as recursively traversing directory structures containing documents. In order to build browsing indexes from metadata, an analogous scheme of "classifiers" is used: classifiers (also written in PERL) create browsing indexes of various kinds 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. To date, collections have been built containing French, Spanish, Maori, Chinese, Arabic and English. 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. Currently it is available in French, German, Spanish, Portuguese, Maori, Chinese, Arabic and English. New languages can be added easily.

Multimedia collections. Greenstone collections can contain text, pictures, audio and even 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 Gutwin *et al.* [3] and Paynter *et al.* [5].

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 [6]. 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. 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 is what you 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

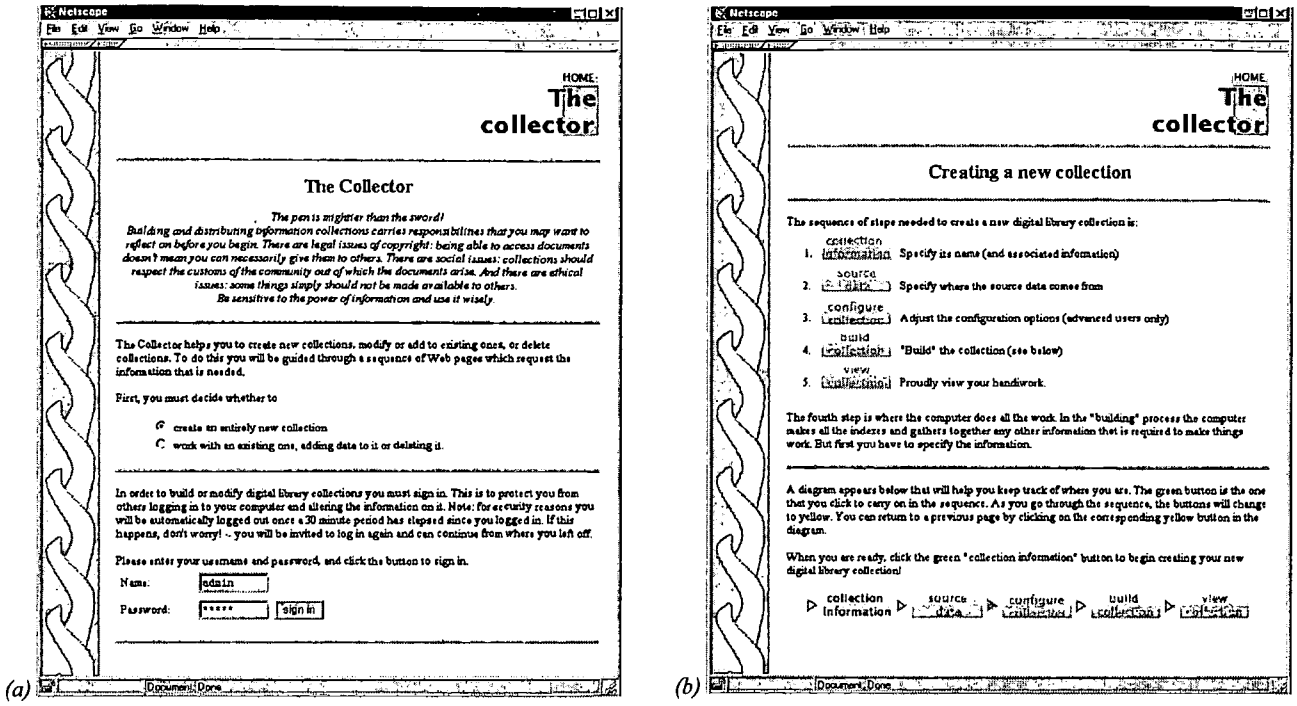


Figure 1 Using the Collector to build a new collection (continued on next pages)

document types. It includes an autoinstall feature to allow easy installation on both Windows and Unix.

3. THE COLLECTOR

Our conception of digital libraries is captured by the following brief characterization [1]:

> *A collection of digital objects, including text, video, and audio, along with methods for access and retrieval, and for selection, organization and maintenance of the collection.*

It is the last point—selection, organization and maintenance of the collection—that we address in this paper. Our view is that just as new books acquired by physical libraries are integrated with the existing catalog on the basis of their metadata, so one should easily be able to add to a digital library without having to edit its content in any way. Furthermore we strive to do this without manual intervention. Once added, such material should immediately become a first-class component of the library. We accomplish this with a build/rebuild process that imports new documents into a library collection using XML to standardize representation, and use explicitly stated metadata to update searching and browsing structures.

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 to it—so long as they have the same format as the existing documents, and the same metadata is provided, in exactly the same way.

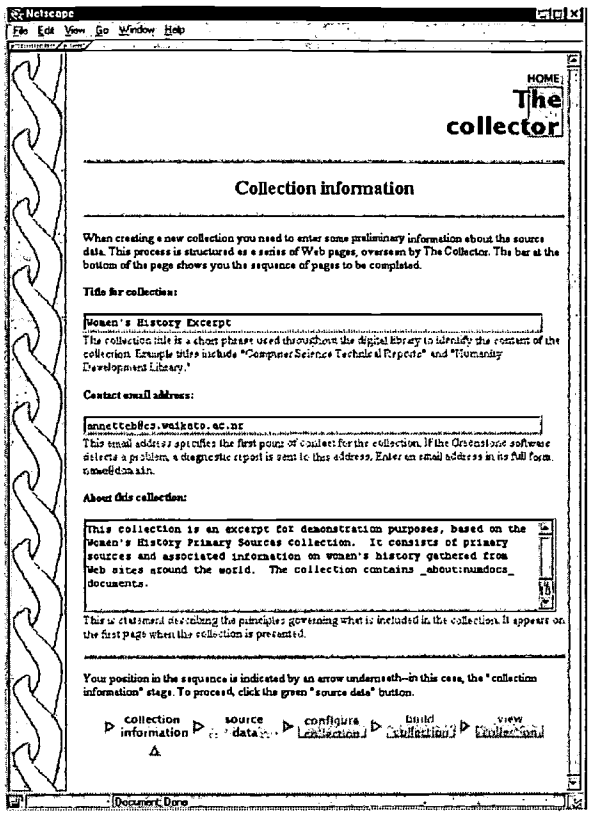
The scheme for collection building has the following basic functions:

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

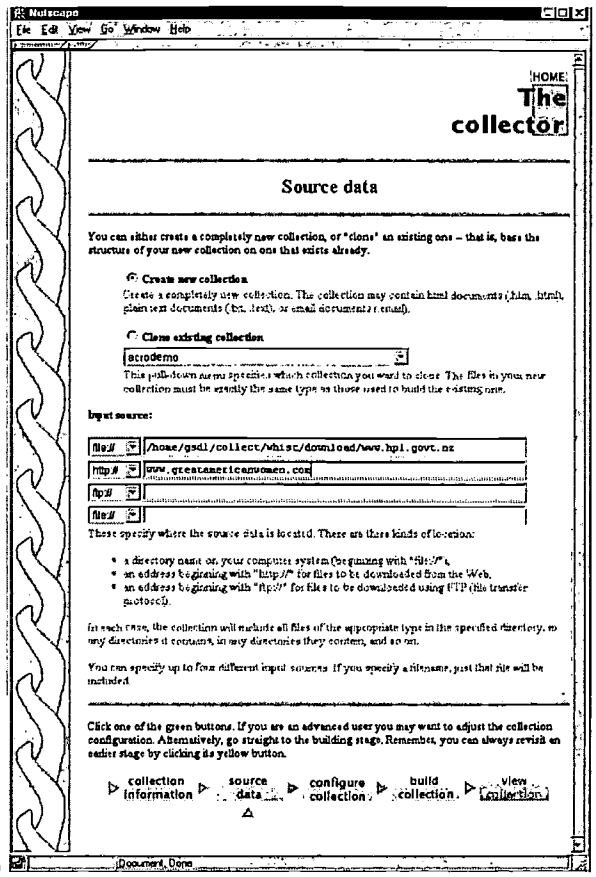
Figure 1 shows the Greenstone Collector being used to create a new collection, in this case from a set of HTML files stored locally. In Figure 1a, the user must first decide whether to work with an existing collection or build a new one. The former case covers the first two options above; the latter covers the remainder. While the example shows a collection being built from existing files, we emphasize that the Collector supports the creation of completely new collections formed from completely new information.

3.1 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 the Greenstone server. Of course, we cannot allow arbitrary people to build collections (for reasons of propriety if nothing else), so Greenstone contains a security system which 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



(c)



(d)

Figure 1 (continued)

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.

3.2 Dialog structure

Upon completion of login, the page in Figure 1b appears. This shows the sequence of steps that are involved in collection building. They are:

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

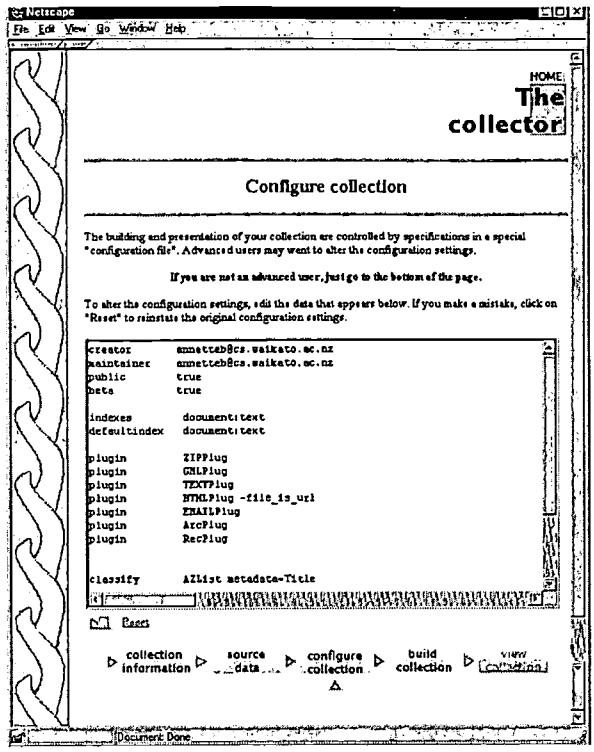
The first step is to specify the collection's name and associated information. The second is to say where the source data is to come from. The third is to adjust the configuration options, which requires considerable understanding of what is going on—it is really for advanced users only. The fourth step is where all the (computer's) work is done. During the "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 gray buttons at the bottom of the screen in Figure 1b, and at the bottom of all other pages 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 (*collection information* in Figure 1b). The gray buttons (all the others, in Figure 1b) are inactive. The buttons change to yellow as you proceed through the sequence, and the user can return to an earlier step by clicking the corresponding yellow button in the diagram. 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.

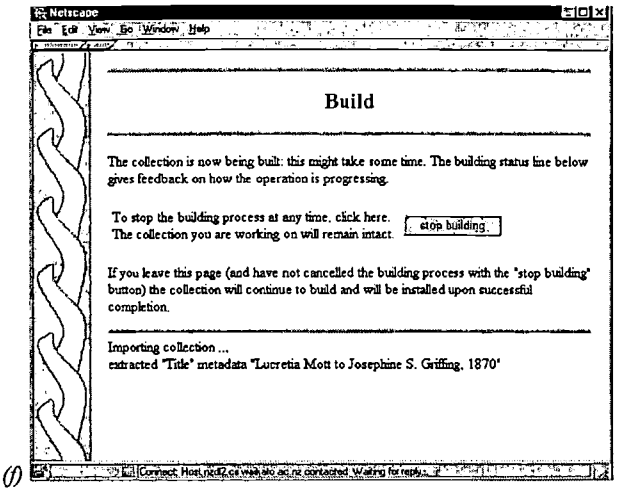
3.3 Collection information

The next step in the sequence, collection information, is shown in Figure 1c. When creating a new collection, it is necessary to enter some information about it:

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



(e)



(f)

Figure 1 (continued)

The collection title is a short phrase used through the digital library to identify the content of the collection: we have already mentioned such titles as *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 describing the principles that govern what is included in the collection. It appears under the heading *About this collection* on the first page when the collection is presented.

Lesk [4] recommends that digital libraries articulate both the principles governing what is included and how the collection is organized. *About this collection* is designed to address the first point. The second is taken care of by the help text, which includes a list of access mechanisms that is automatically generated by the system based on what searching and browsing facilities are included in the collection.

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, as Figure 1c shows, the *collection information* stage. The user proceeds to Figure 1d by clicking the green *source data* button.

3.4 Source data

Figure 1d is the point where 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 is a major undertaking, and is not what the interactive Collector interface is designed for. The most effective way to create a new collection is 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,[2] there is a good chance that a suitable structure exists. It is preferable that the document file types in the new collection are amongst those catered for by the old one, the same metadata is available, and the metadata is 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 simply 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 in the latter case it emanates from 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: up to four separate input sources can be specified. There are three kinds of specification:

[2] Collections can be downloaded from nzdl.org.

```
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 2. Configuration file for collection generated in Figure 1

- 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.

In the given example (Figure 1d) the new collection will contain documents taken from a local file system as well as a remote Web site, which will be mirrored during the building process, thus forming a new resource that is the composite of the two.

3.5 Configuring the collection

Figure 1e shows the next stage. The construction and presentation of all collections is controlled by specifications in a special collection configuration file (see below). Advanced users may use this page to alter the configuration settings. Most, however, will proceed directly to the final stage.

In the given example the user has made a small modification to the default configuration file by including the *file_is_url* flag with the HTML plugin. This flag causes URL metadata to be inserted in each document, based on the filename convention that is adopted by the mirroring package. This metadata is used in the collection to allow readers to refer to the original source material, rather than to a local copy.

3.6 Building the collection

Figure 1f shows the "building" stage. 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 for it 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 (we use *wget*[3]) is used to recursively copy the specified site along with any related image files.

Next, the documents are converted into XML. 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 the user can stop the building process

[3] See www.gnu.org

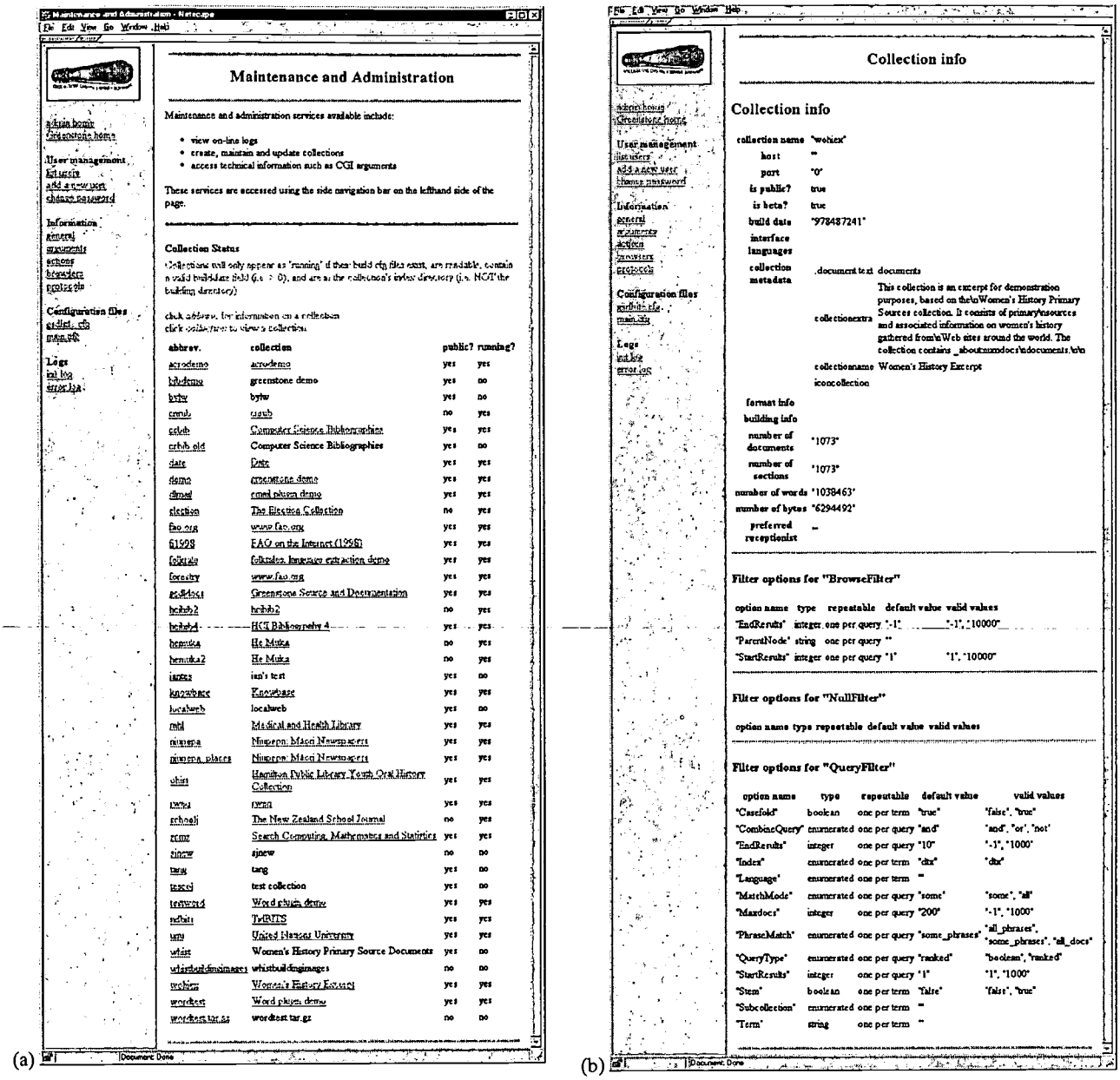


Figure 3 The Greenstone Administration facility

immediately using the button in Figure 1f, there is no reliable way to prevent users 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 the status area at the bottom part of Figure 1f, updated every five seconds. The message visible in Figure 1f indicates that when the snapshot was taken, *Title* metadata was being extracted from an input file. 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.

3.7 Viewing the collection

When the collection is built and installed, the sequence of buttons visible at the bottom of Figures 1a–e appears at the bottom of Figure 1f, with the *View collection* button 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 system's 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.

3.8 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 dialog structure illustrated in Figure 1 is used: entry is at the "source data" stage, Figure 1d. The new data that is specified is copied as before and converted to GML, joining any existing imported material.

Revisions of old documents should perhaps replace them 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. Recall, the aim of the Collector is to support the most common tasks in a straightforward manner—more careful updating is possible through the command line.

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 in Figure 1e.

Deleting a collection simply requires a collection to be selected from a list, and its deletion confirmed. 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 the necessary files are automatically massaged into a disk image in a standard directory.

4. THE COLLECTION CONFIGURATION FILE

Part of the collection configuration file for the collection built in Figure 1 is shown in Figure 1e; it appears in full in Figure 2. Since we are in the process of updating the collection configuration file format to support a wider variety of services, we will not embark on a detailed explanation of what each line means.

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. In essence this is "collection-level metadata" and we are studying existing standards for expressing such information. 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 will uncompress any Zipped files; 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—stored in an XML format—will be processed properly when the collection is rebuilt. 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.

More indicative of Greenstone's power than the generic structure in Figure 2 is the ease with which other facilities can be added. To choose just ten examples:

- A full-text-searchable index of titles could be added with one addition to the *indexes* line.
- If authors' names were encoded in the Web pages using the HTML metaname construct, a corresponding index of authors could also be added by augmenting the *indexes* line.
- With author metadata, an alphabetic author browser would require one 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 [9] and a list of acronyms added
- Keyphrases could be extracted from each document [2] and a keyphrase browser added
- A phrase hierarchy could be extracted from the full text of the documents and made available for browsing [5]
- 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 panel in Figure 1e. However, we do not anticipate that casual users will operate at this level, and provision is made in the Collector to by-pass this editing step. 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.

5. SUPPORT FOR THE SYSTEM ADMINISTRATOR

An "administrative" facility is included with every Greenstone installation. The entry page, shown in Figure 3a, gives information about each of the collections offered by the system. Note that *all* collections are included—for there may be "private" ones that do not appear on the Greenstone home page. With each is given its short name, full name, whether it is publicly displayed, and whether or not it is running. Clicking a particular collection's abbreviation (the first column of links in Figure 3a) brings up information about that collection, gathered from its collection configuration file and from other internal structures created for that collection. If the collection is both public and running, clicking the collection's full name (the second link) takes you to the collection itself.

The collection we have just built has been named *wohiex*, for *Women's History Excerpt*, and is visible near the bottom of Figure 3a. Figure 3b shows the information that is displayed when this link is clicked. The first section gives some information from the configuration file, and the size of the collection (1000 documents, a million words, over 6 Mb). The next sections contain internal information related to the communication protocol through which collections are accessed. For example, the filter options for "QueryFilter" show the options and possible values that can be used when querying the collection.

The administrative facility also presents configuration information about the installation and allows it to be modified. It facilitates examination of the error logs that record internal errors, and the user logs that record usage. It enables a specified user (or users) to authorize others to build collections and add new material to existing ones. All these facilities are accessed interactively from the menu items at the left-hand side of Figure 3a.

5.1 Configuration files

There are two configuration files that control Greenstone's overall operation: the site configuration file *gsdlsite.cfg*, and the main configuration file *main.cfg*. The former is used to configure the Greenstone software for the site where it is installed. It is designed for keeping configuration options that are particular to a given site. Examples include the name of the directory where the Greenstone software is kept, the HTTP address of the Greenstone system, and whether the *fastcgi* facility is being used. The latter contains information that is common to the interface of all collections served by a Greenstone site. It includes the email address of the system maintainer, whether the status and collector pages are enabled, and whether cookies are used to identify users.

5.2 Logs

Three kinds of logs can be examined: usage logs, error logs and initialization logs. The last two are only really of interest to people maintaining the software. All user activity—every page that each user visits—can be recorded by the Greenstone software, though no personal names are included in the logs. Logging, disabled by default, is enabled by including an appropriate instruction in the main system configuration file.

Each line in the user log records a page visited—even the pages generated to inspect the log files! It contains (a) the IP address of the user's computer, (b) a timestamp in square brackets, (c) the CGI arguments in parentheses, and (d) the name of the user's browser (Netscape is called "Mozilla"). Here is a sample line, split and annotated for ease of reading:

```
   /fast-cgi-bin/niupepalibrary
(a) its-www1.massey.ac.nz
(b) [950647983]
(c) (a=p, b=0, bcp=, beu=, c=niupepa, cc=, ccp=0, ccs=0,
      cl=, cm=, cq2=, d=, e=, er=, f=0, fc=1, gc=0,
      gg=text, gt=0, h=, h2=, hl=1, hp=, il=1, j=, j2=,
      k=1, ky=, l=en, m=50, n=, n2=, o=20, p=home, pw=,
      q=, q2=, r=1, s=0, sp=frameset, t=1, ua=, uan=,
      ug=, uma=listusers, umc=, umnpw1=, umnpw2=, umpw=,
      umug=, umun=, umus=, un=, us=invalid, v=0, w=w,
      x=0, z=130.123.128.4-950647871)
(d) "Mozilla/4.08 [en] (Win95; I ;Nav)"
```

The last CGI argument, "z", is an identification code or "cookie" generated by the user's browser: it comprises the user's IP number followed by the timestamp when they first accessed the digital library.

5.3 User authentication

Greenstone incorporates an authentication scheme that can be employed to control access to certain facilities. It is used, for example, to restrict the people who are allowed to enter the Collector and certain administration functions. It also allows documents to be protected on an individual basis so that they can only be accessed by registered users on presentation of a password; however this is currently cumbersome to use and needs to be developed further. Authentication is done by requesting a user name and password as illustrated in Figure 1a.

From the administration page users can be listed, new ones added, and old ones deleted. The ability to do this is of course also protected: only users who have administrative privileges can add new users. It is also possible for each user to belong to different "groups". At present, the only extant groups are "administrator" and "colbuilder". Members of the first group can add and remove users, and change their groups. Members of the second can access the facilities described above to build new collections and alter (and delete) existing ones.

When Greenstone is installed, there is one user called *admin* who belongs to both groups. The password for this user is set during the installation process. This user can create new names and passwords for users who belong just to the *colbuilder* group, which is the recommended way of giving other users the ability to build collections.

User information is recorded in two databases that are placed in the Greenstone file structure. One contains all information relating to users. The other contains temporary keys that are created for each page access, which expire after half an hour. Thus inactive users must reauthenticate themselves.

5.4 Technical information

The links under the *Technical information* heading gives access to more technical information on the installation, including the directories where things are stored.

6. User Evaluation

The Collector and administration pages have been produced and refined through a long period of iterative design and informal testing. The design underwent many revisions before reaching the version presented in this paper. The details were thrashed out over several meetings of our digital library group—some 20 or so individuals from a variety of disciplines including library science, the humanities, and notably within computer science, the field of human computer interaction. It was here, for example, that the idea for the progress bar at the bottom of the page was formulated, and the very name of the tool, the Collector, was conceived. Once satisfied with its development, the tool was added into the public release of the Greenstone software.

Further feedback was obtained through the Greenstone mailing list, a general purpose listserver for Greenstone. In this arena both current issues and future developments are discussed, and users can gain technical assistance for particular problems. Filtering this source specifically for remarks about the Collector and administration pages revealed only technical questions—normally connected, despite extensive prior testing, with scripts performing

incorrectly on a given version of a particular operating system. None of the questions fell into the category "how do I do this?" with the Collector. The technical questions indicate that the tools are being used, and the dearth of "how to" questions suggests that it is performing adequately.

Members of our group are presently conducting a usability study designed to establish more clearly how suitable the interface is for performing its intended tasks. Volunteers from a final year computer science class are observed performing set tasks: using both the tools described here and the original set of command line instructions. On completion of the tasks, which take about an hour, the users then fill out a questionnaire. The results from this work are not yet available.

The main difficulty of formal usability testing is that it is inevitably artificial. The tasks are artificial, the work environment is artificial, even the motivation of the users is artificial. Furthermore, the parameters of operation are far more tightly prescribed than in the real world. The question posed is how well does the tool let users perform the tasks it was designed for. But what if the user wants to step outside of the design parameters? On the one hand, an unfair question—the Collector is designed to simplify commonly executed tasks. But as users become more familiar with a tool they naturally expect more from it.

There questions are open ended, but vitally important. For their resolution we look to the open source nature of the Greenstone project. We will continue iteratively developing these tools based on feedback from field trials and user comments. We will also incorporate features added by others who actively develop the Greenstone code.

7. CONCLUSIONS

This paper has described the Collector, a tool integrated into the Greenstone environment that guides an end user through building and maintaining a collection, step by step. It also details the support included for the overall administration and maintenance of a Greenstone site by the system administrator, which is likewise integrated into the runtime system and gives the ability to view logs, create new users and control the access they have to, for example, the collection building tool.

Different users of digital libraries, naturally, have different needs. While access and retrieval is an obvious requirement, and dominates digital library research, we believe that end-user collection creation is another important element that deserves careful attention and further development. Including this capability in digital library systems will help them move away from the large and mostly static entities currently seen, and evolve into more dynamic and responsive environments.

8. REFERENCES

[1] Akscyn, R.M. and Witten, I.H. (1998) "Report on First Summit on International Cooperation on Digital Libraries." ks.com/idla-wp-oct98.

[2] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. and Nevill-Manning, C. (1999) "Domain-specific keyphrase extraction." *Proc Int Joint Conference on Artificial Intelligence*, Stockholm, Sweden. San Francisco, CA: Morgan Kaufmann Publishers, pp. 668–673.

[3] 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* 27(1/2): 81–104; November.

[4] Lesk, M. (1997) *Practical digital libraries*. San Francisco, CA: Morgan Kaufmann.

[5] 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.

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

[7] Witten, I.H., Loots, M., Trujillo, M.F. and Bainbridge, D. (2000) "The promise of digital libraries in developing countries."

[8] Witten, I.H., McNab, R.J., Boddie, S.J. and Bainbridge, D. (2000) "Greenstone: A comprehensive open-source digital library software system." *Proc Digital Libraries 2000*, San Antonio, Texas, pp. 113–121.

[9] Yeates, S., Bainbridge, D. and Witten, I.H. (2000) "Using compression to identify acronyms in text." *Proc Data Compression Conference*, edited by J.A. Storer and M. Cohn. IEEE Press Los Alamitos, CA, p. 582. (Full version available as Working Paper 00/1, Department of Computer Science, University of Waikato.)

U.S. Department of Education
Office of Educatonal Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

REPRODUCTION RELEASE
(Specific Document)

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").