# Assembling and Enriching Digital Library Collections

David Bainbridge, John Thompson and Ian H. Witten
*Department of Computer Science*
*University of Waikato*
*Hamilton, New Zealand*
*{davidb, jmt12, ihw}@cs.waikato.ac.nz*

## Abstract

*People who create digital libraries need to gather together the raw material, add metadata as necessary, and design and build new collections. This paper sets out the requirements for these tasks and describes a new tool that supports them interactively, making it easy for users to create their own collections from electronic files of all types. The process involves selecting documents for inclusion, coming up with a suitable metadata set, assigning metadata to each document or group of documents, designing the form of the collection in terms of document formats, searchable indexes, and browsing facilities, building the necessary indexes and data structures, and putting the collection in place for others to use. Moreover, different situations require different workflows, and the system must be flexible enough to cope with these demands. Although the tool is specific to the Greenstone digital library software, the underlying ideas should prove useful in more general contexts.*

## 1. Introduction

One attractive feature of digital libraries is that they make it possible—even easy—to assemble new anthologies of targeted information, organize them, provide worldwide access through the Internet, and distribute them on removable media. Moreover, this can be accomplished by domain specialists who do not necessarily have information science training and who lack the institutional backup of a conventional library catalog room.

These advantages could be more widely realized if appropriate support were available for people who create information collections. Such people, who have been called "corpus editors" [6], need an interactive tool that facilitates the entire process of building digital library collections, and that is what this paper describes. We assume that the source material is available electronically, or has already been scanned into digital form. In practice when building digital library collections it is often necessary to deal with paper documents and the process of digitization and OCR [10]; however, we cannot address these issues here.

The requirement for such a tool originated in the context of digital libraries for sustainable development [13]. Effective human development blossoms from empowerment rather than gifting: as the Chinese proverb says, "Give a man a fish and he will eat for a day; teach him to fish and he will eat for the rest of his days." Although disseminating information originating in the developed world is certainly a useful activity, a more effective strategy for sustained long-term human development is to disseminate the capability of creating information collections rather than the collections themselves. And if we can help inhabitants of developing countries to assemble their own information collections, surely we can use the same techniques in our own homes and institutions too.

The tool that we describe here, called the "Gatherer," works in consort with the Greenstone digital library software [14, 16], and is GPL-licensed open source software. This makes the work of building it much easier, for other GPL modules can be incorporated verbatim—and we have taken full advantage of this. It is implemented in Java, for portability and efficiency, and uses the Swing user interface system [4].

This paper begins by identifying the requirements for assembling and enriching digital library collections, and then reviews public domain digital library software tools that help with collection creation. We next describe the Gatherer by presenting a detailed illustrated walkthrough of its operation and then expand upon this to describe how it supports different workflows.

## 1. Requirements

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—that is addressed 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 material to a digital library without having to edit its content in any

Table 1 Implementation summary of sample open source digital library systems

| System | URL | Platform | Technology | Supported standards |
|---|---|---|---|---|
| CDS/ISIS | www.unesco.org/webworld/isis | Windows | Various | |
| Harvest | harvest.sourceforge.net | Unix | C, PERL, Bison, Flex | |
| Koha | www.Koha.org | Unix | mySQL, PERL | Z39.50, MARC |
| EPrints | www.EPrints.org | Unix | mySQL, PERL | Open Archives |
| DSpace | www.DSpace.org | Java/Unix | postgreSQL, Lucene (indexing) | Open Archives |
| Greenstone | www.greenstone.org | Unix, Windows, MaxOS X | GDBM, MG (indexing), PERL (building), C++ (runtime) | Z39.50, Open Archives , MARC |

way. Once added, such material should immediately become a first-class component of the library, fully integrated with existing search and browsing structures through explicitly-stated metadata. The challenge is to provide an easy-to-use interactive interface to help librarians provide appropriate metadata, and design and build collections.

The first step in putting together a new information collection is to gather the source material from local files, or from the Web, or perhaps from already-existing digital library collections. These files may be in many different formats. Furthermore, they may include metadata, perhaps contained in document files, perhaps in separate metadata files, or perhaps already embedded in an existing digital library collection.

Next, it is necessary to select, define, or modify a metadata set. When importing existing metadata, it may be necessary to convert between different standards, or modify an existing metadata schema to include new elements. Also, it is frequently convenient to be able to specify metadata for each document interactively. Sometimes one wishes to assign the same metadata value to a group of files, or directories, in a single operation. New element values should be stored so that they can be re-used by selecting them, to reduce the risk of introducing errors through retyping.

The next step is to design the collection structure around the available metadata, build the indexes and browsing structures, and move the collection to a place where it can be served to potential users. These operations tend to be specific to a particular digital library system. In our case, we have used Greenstone as the underlying digital library infrastructure.

The extensible nature of open-source software poses its own difficulties. New modules ("plug-ins" and "classifiers" in Greenstone terminology; see below) are self-documenting in that they contain information on the options and switches they support. This needs to be garnered automatically and presented to users where appropriate when they are designing collections.

Building digital library collections inevitably involves some consideration of workflow. Is the collection built on a local computer or a central digital library server? If the latter, do the files originate on the server or the local workstation from which the digital library is being accessed? Or is the material sent to a central collection editor from one or more remote locations? And if so, where is the metadata assigned—remotely, centrally, a combination of both, or produced remotely and checked and edited centrally?

On the whole, existing systems assume a specific workflow model that is motivated by the needs of the intended user base. However, in our work we strive for greater generality. A range of different workflows can be accommodated, and we believe that the design is flexible enough to be used in situations that have not explicitly been envisaged. For example, in the "remote submission/central collection" scenario, our system helps in both roles: preparing metadata that accompanies documents when they are submitted, and reviewing and editing submissions before incorporating them into the collection.

## 1. Related Work

Recent years have seen a steep rise in the number of digital library solutions available, and it is impractical to review them all here. Instead we concentrate on examining pertinent strategies that have been successfully deployed. Proprietary systems are not discussed because insufficient design information is available.

The designers of the digital library systems described below generally use off-the-shelf technology to implement significant parts of the system. In particular they use a database (typically relational) to store and retrieve metadata, and an indexing tool if full-text searching of document content is to be provided. A web server capable of running CGI scripts is another widely used component, because the most prevalent form of digital library user interface acts through a web browser. These components are bound together by custom software that implements the workflow constituting the digital library system.

Table 1 lists six open source digital library systems, the last being our own, and summarizes the platforms, principal implementation technologies, and component standards.

CDS/ISIS is an information storage and retrieval system that has been under continual development by

UNESCO since 1985 and is intended, in particular, to address the requirements of developing countries [3]. It provides a multilingual text database that is widely used for storing bibliographic information and distributing it on CD-ROM, and also for thesaurus management.

Harvest is a system designed to collect information and make it searchable through a web interface [7]. It has a module (also called "Gatherer," but no relation to the system described here) that extracts metadata from source documents using a *flex* parser for pattern matching. Customizing extraction is, in essence, writing a new set of *flex* rules. Extracted metadata is used as a "snippet" to present to the user when displaying search results. However, it is not possible to restrict searches to particular metadata fields.

EPrints is designed for archiving on-line reports, particularly academic research papers [8, 9]. Indexing and a subject hierarchy are based around a relational database of bibliographic metadata that a site administrator can adjust to suit the sort of reports being stored. Source documents are linked to their corresponding bibliographic record so that they can be accessed at runtime as a side-product of record retrieval. Through a network of web pages, end-users (who are often authors of papers in the archive) and the administrator participate in a workflow cycle of submission (article and descriptive metadata), optional editing control, and deposit—thereby accomplishing the developers' stated aim of a "self-archiving" methodology.

Using exactly the same implementation technology, Koha[1] focuses on a different community—public librarians and their users [2]. The services provided are strikingly different to those offered by EPrints, but the underlying techniques are similar. In addition to bibliographic information (expressed in MARC format) in the relational database, recorded items are extended to include membership, reading list, acquisition, and budget information, amongst other things. However, the same basic arrangement of web forms is used to support a workflow of submission and editing (password protected if required). From the user's perspective, Koha is like the graphical, web-based OPAC systems one has come to associate with public libraries.

DSpace, a combined venture by MIT and Hewlett Packard, targets the digital content needs of institutions [17]. Strongly influenced by the Open Archival Information Systems (OAIS) reference model [5], DSpace supports submission, searching, browsing and retrieval. Searching includes full-text retrieval, which, through the use of Apache's Lucene indexing tool, allows for incremental updates.

Considerable attention is paid to the submission workflow. Users wishing to submit items first log into DSpace, then enter files and associate metadata with them using a modified Dublin Core schema. On receipt of a submission, DSpace notifies the following people:

*Reviewer*, who accepts or rejects the submission
*Approver*, who checks for file errors, etc.
*Editor*, who checks and augments metadata.

Once these steps are complete, the submission is archived into the system, paying particular attention to the data format and level of preservation.

## 1.1 Greenstone

Based on the study of the requirements, we have designed a new piece of software, the Gatherer, which supports interactive collection building and is closely coupled with Greenstone. The design has benefited greatly from our study of other open source digital library systems, and from our own experience with two existing Greenstone modules, the Collector [15] and the Organizer [12]. We have also learned from the "Metadata Editor," an independent software module created by a separate organization that works with Greenstone. We describe these related systems first before providing an account of the Gatherer in the next section.

Developed over the last five years, the Greenstone open source digital library software from the New Zealand Digital Library project enjoys considerable success and is widely used [14, 16]. It 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. Its strengths include international language support, multilingual interfaces, and a flexible document importing process that handles different formats—HTML, Word, PDF, PostScript, and E-mail messages, to name but a few. Images, video and audio require accompanying textual metadata.

For the purposes of the present paper it is necessary to learn a little about three key design features of Greenstone: plug-ins, classifiers, and the collection configuration file. In Greenstone, the structure, organization, and presentation of any particular collection are 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 sources of metadata, what browsing facilities are to be provided, what full-text search indexes are required, 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 structure, organization, and presentation of the collection is recorded in a text file called the "collection configuration file."

Source material is imported into the system through "plug-ins" that cater for different document formats. Any given collection may have source documents in many

[1] The name is a Maori word meaning gift or donation, chosen to reflect the open source nature of this project.

different forms. There are plug-ins to handle all the file formats mentioned above, along with some proprietary formats, and for generic tasks such as recursively traversing directory structures containing such documents. Plug-ins can be written to accommodate new document types.

Greenstone builds browsing indexes from metadata. This is done using "classifiers," which are analogous to plug-ins. Classifiers create browsing indexes of various kinds, based on metadata—alphabetically tabbed lists (of titles, for example), date-selected lists, and hierarchical browsing structures. Like plug-ins, new classifiers can be written for special-purpose browsing structures.

## 1.1 The Collector

Part of Greenstone, the Collector is a utility for collection building with the following basic functions [15]:

- 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;
- write an existing collection to a self-contained, self-installing CD-ROM.

It is modeled after popular end-user installation software (such as InstallShield[2]). 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 clearly.

The Collector is implemented as a web-based system. A user goes to a certain web address in any Greenstone installation to find a page that facilitates collection building. The ability to build collections is protected: users must log in first, and only those with appropriate privileges are allowed to build collections. Being web-based imposes certain restrictions. First, there are security issues with letting remote users transfer files to the site running the web server. Second, the user's local file space is only available in a limited way through the file-upload mechanism, meaning that only one file at a time can be specified. Third, web-based interaction was not designed for lengthy processes such as building digital library collections (which can take many hours of processor time for non-trivial collections), and special measures must be taken to ensure that the user receives appropriate feedback, and that the right things happen on moving to another page, or pressing the *Back* button, while the collection is being built.

Greenstone is so easy to install and run that, contrary to the assumptions we made when designing the Collector, users often work on a local copy of the software that runs on their own computer. They quite reasonably expect there to be a convenient mechanism for populating their collections with local files, which the Collector cannot do directly.

## 1.1 The Organizer

One of Greenstone's early adopters, SimpleWords of Brasov, Romania, uses it to build and distribute collections for humanitarian purposes—collections that share a common collection configuration file but have different content. Librarians must manually assign a substantial amount of metadata to each document in the collection. They do this using a uniform metadata scheme called the Development Library Set (DLS). However, the Collector does not provide any interactive way of specifying metadata to be added to documents. Although Greenstone accommodates different metadata formats, including a simple but flexible XML format, OAI, MARC records, a spreadsheet format, BibTeX, and even Refer, all metadata must be available in advance, stored in files.

SimpleWords implemented a separate utility, the Organizer, to create and edit the material associated with a collection conforming to their humanitarian format [12]. This is a Microsoft Visual C++ application, distributed with Greenstone. It assists in the creation of collections that use DLS.

The Organizer is designed to help manage all aspects of organizing a digital library collection: entering document titles, assigning subjects and other metadata, altering them, etc. It is, however, limited to a particular document model, and users who wish to add metadata not included in this model would have to make manual modifications to the Organizer's results. Also, because it is restricted to Windows whereas Greenstone runs under Unix and MacOS X too, the Organizer is not fully integrated but works by generating intermediate files that Greenstone uses when building collections. While this loose coupling has some advantages in flexibility, it prevents the full collection-building process from being a single, integrated, task.

## 1.1 The Metadata Creator

Another organization, the Mercy Corps, centered in Portland Oregon and with operations in about thirty of the world's most unstable countries, uses Greenstone to organize its extensive collection of in-house documents, manuals, forms, and memos. They have made significant enhancements to support a workflow for new acquisitions to the library.

In their organization, field offices submit new documents by filling out metadata using a simple web-based form and attaching the document. This arrives in the in-tray of a central librarian who checks it for correctness and integrity before including it in the appropriate collection. Collections, rebuilt automatically every night, are available on the web for in-house use,

[2] www.installshield.com

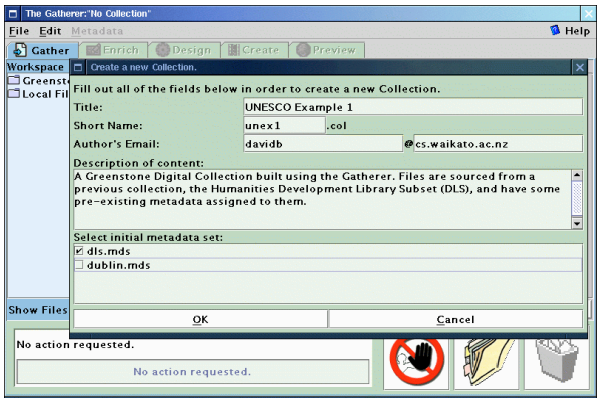


Figure 1. Starting a new collection

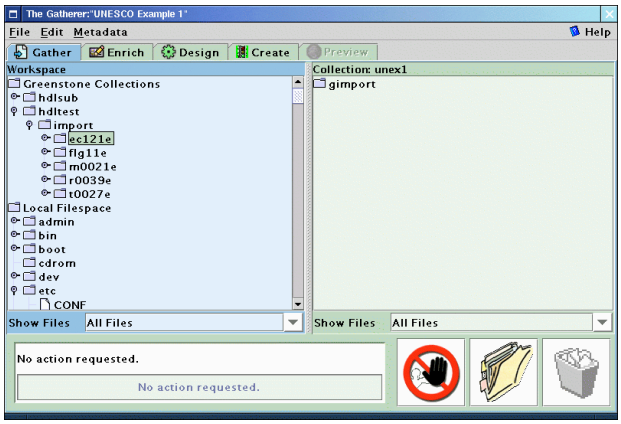


Figure 2. Exploring the local file space

and are written at regular intervals to CD-ROM for physical distribution.

Mercy Corps designed and implemented the Metadata Creator, an interactive web-based database system that allows users to enter and edit the metadata associated with a document. This module is used both by people in the field office when initially entering metadata, and by the central librarian who checks it. For the former it incorporates facilities for automatically submitting the document and metadata to the central office, while for the latter it makes it easy to periodically rebuild the Greenstone collections with the new documents.

Although the Metadata Creator is at present restricted to the Mercy Corps' own in-house metadata standard, they are planning to release a Dublin Core version to the Greenstone community for general usage. Though it will remain useful as a lightweight metadata entry utility, like the Organizer it does not provide a tight coupling with the collection-building process, a serious omission from our requirements analysis.

## 1. The Gatherer: Functionality

The Gatherer allows users to collect sets of documents, import or assign metadata, and build them into a Greenstone collection. It differs from the Collector in that its default behaviour is to work on the computer running the Greenstone digital library software rather than building collections on a remote machine, and this permits a more flexible interface. It differs from the Organizer and the Metadata Creator in that it deals with unrestricted metadata sets, and can be tightly integrated with the Greenstone collection design and creation process.

The Gatherer is written in the Java language and is platform-independent. It incorporates various open-source packages for such tasks as file browsing, HTML rendering, web mirroring, and efficient table sorting.

The user works with it to build a Greenstone collection. It supports five basic activities, which can be interleaved but are nominally undertaken in this order:

1. Copy documents from the computer's file space, including existing collections, into the new collection. Any existing metadata remains "attached" to these documents. Documents may also be gathered from the web through a built-in mirroring facility.
1. Enrich the documents by adding further metadata to individual documents or groups of documents.
1. Design the collection by determining its appearance and the access facilities that it will support.
1. Build the collection using Greenstone.
1. Preview the newly created collection.

### 1.1 Assembling the source material

To convey the operation of the Gatherer we work through a simple example. Figures 1 to 12 are screen snapshots at various points during the interaction. This example uses documents in the Humanity Development Library Subset collection, which is distributed with Greenstone. For expository purposes, the walkthrough takes the form of a single pass through the steps listed above. A more realistic pattern of use, however, is for users to switch back and forth through the various stages as the task proceeds.

Since Greenstone is a lightweight system that can easily be installed on small workstations or laptops, we have decided to focus on the situation where the collection editor works interactively on the same computer where collections are built. However, it is possible to use the Gatherer to collect document files and define interaction locally, without a digital library installation, and send them to a central location for building into a collection and serving on the Internet. At the central location a librarian can also use the Gatherer when deciding whether the documents are suitable to be added to the collection, and to check and edit the metadata that has been associated with them. This affords a degree of flexibility for different workflow requirements.

To commence interaction with the panels, the user must either open an existing collection or begin a new

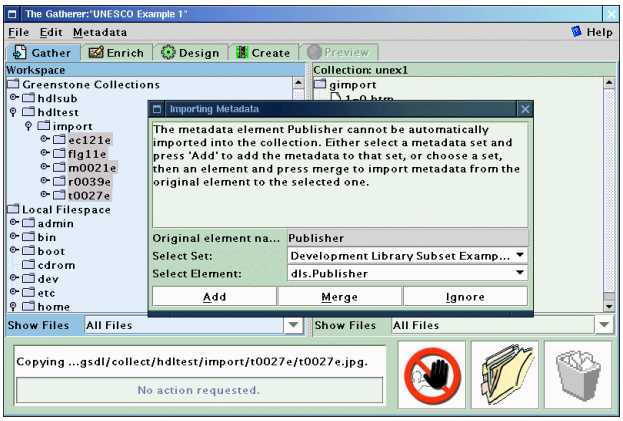


Figure 3. Importing existing metadata

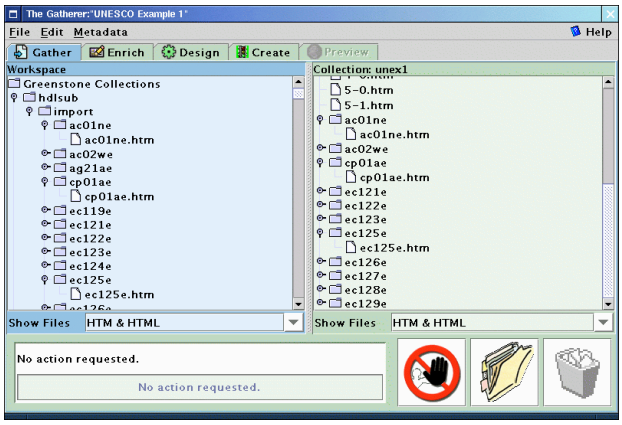


Figure 4. Filtering the file trees

one. Figure 1 shows the user in the process of starting a new collection. She has selected *New* from the file menu and begun to fill out general information about the collection—its name, the E-mail address of the person responsible for it, and a brief description of the content—in the popup window. The collection name is a short phrase used throughout the digital library to identify the collection's content: existing collections have names like *Food and Nutrition Library*, *World Environmental Library*, and so on. The E-mail 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.

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 collection's initial page. Lesk [11] 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. Greenstone takes care of the second using help text, which is formed by automatically generating a list of access mechanisms based on the searching and browsing facilities that are included in the collection.

At this point, a metadata set is selected. Dublin Core is pre-supplied, but the user can create new metadata using a popup panel activated through the "metadata" menu. This has already been done for the DLS metadata set mentioned above, and the user has chosen this item for her new collection. Several different metadata sets can be associated with the same collection; the system keeps them distinct (so that, for example, documents can have both a Dublin Core *Title* and a DLS *Title*). Behind the scenes, metadata sets are represented in XML.

After clicking the *OK* button on the "new collection" popup, the remaining parts of the interface, which were grayed out before, become active. The *Gather* panel, selected by the eponymous tab near the top of Figure 1, is displayed initially. This allows the user to explore the local file space and existing collections, gathering up selected documents for the new collection. The panel is divided into two sections, the left for browsing existing structures and the right for the documents in the collection.

Operations available at this stage include:

- Navigating the existing file structure hierarchy, and the one being created, in the usual way.
- Dragging and dropping files into the new collection.
- Multiple selection of files.
- Dragging and dropping entire sub-hierarchies.
- Deleting documents from the nascent collection.
- Creating new sub-hierarchies within the collection.
- Filtering the files that are visible, in both the local file system and the collection, based on predetermined groups or on standard file matching terms.
- Invoking the appropriate program to display the contents of a selected file, by double-clicking it.

Care is taken to deal appropriately with name clashes when files of the same name in different parts of the computer's directory structure are copied into the same folder of the collection.

In Figure 2 the user is using the interactive file tree display to explore the local file system. At this stage, the collection on the right is empty; the user populates it by dragging and dropping files of interest from the left to the right panel. Such files are "copied" rather than "moved": so as not to disturb the original file system. The usual techniques for multiple selection, dragging and dropping, structuring the new collection by creating subdirectories ("folders"), and deleting files from it by moving them to a trashcan, are all available. Through an extra panel, which can optionally be activated, the user can browse external Web sites and selectively mirror them; the result of this copying appears as another top-level folder on the left-hand side of the *Gather* panel.

Existing collections are represented by a subdirectory on the left called "Greenstone Collections," which can be opened and explored like any other directory. However, the documents therein differ from ordinary files because they already have metadata attached, which the Gatherer preserves when it moves them into the new collection.

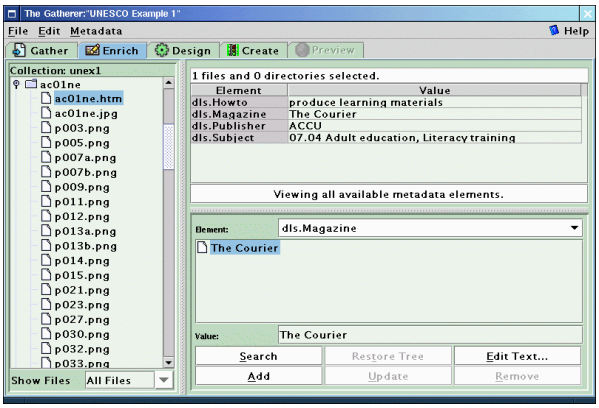


Figure 5. Assigning metadata using the *Enrich* view

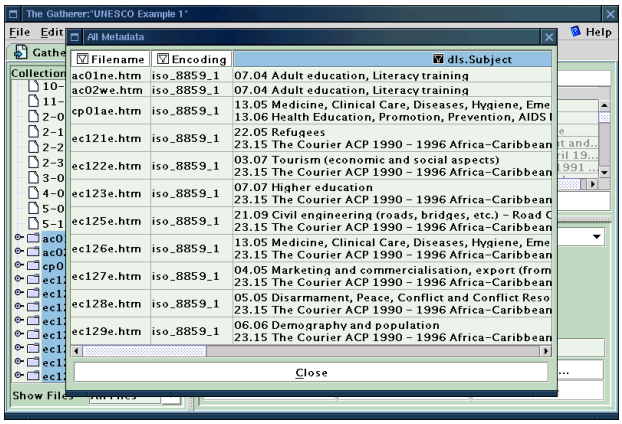


Figure 6. Viewing all metadata assigned to selected files

Conflicts may arise because their metadata may have been assigned using a different metadata set from the one in use for the new collection, and the Gatherer helps the user resolve these. In Figure 3 the user has selected some documents from an existing collection and dragged them into the new one. The popup window explains that the metadata element *Publisher* cannot be automatically imported, and asks the user to either select a metadata set and press *Add* to add the metadata element to that set, or choose a metadata set, then an element, and press *Merge* to effectively rename the old metadata element to the new one by merging the two. Metadata in subsequent documents will automatically be handled in the same way.

When large file sets are selected, dragged, and dropped into the new collection, the copying operation may take some time—particularly if metadata conversion is involved. To indicate progress, the Gatherer shows which file is being copied and what percentage of files has been processed. The implementation is multi-threaded: the user can proceed to another stage while copying is still in progress.

Special facilities are needed for dealing with large file sets. For example, the user can choose to filter the file tree to show only certain files, using a dropdown menu of file types displayed underneath the trees. In Figure 4, only the HTM and HTML files are being shown (and only these files will be copied by drag and drop).

Another source of documents is the web itself. The Gatherer has a *Mirror* panel that allows the user to interact with a mini web browser and select certain pages, or sites, for mirroring. There are many options: mirroring depth, automatically download embedded objects like images, only mirror from the same site, etc. The actual download operation is accomplished by *wget*, a widely-used open-source mirroring utility.

## 1.2 Enriching the documents

The next phase in collection building is to enrich the documents by adding metadata. The *Enrich* tab brings up a new panel of information (Figure 5), which shows the document tree representing the collection on the left and on the right allows metadata to be added to individual documents, or groups of documents.

Documents that are copied during the first step come with any applicable metadata attached. If a document is part of a Greenstone collection, previously defined metadata is carried over to the new collection. Of course, this new collection may have a different metadata set, or perhaps just a subset of the defined metadata, and only metadata that pertains to the new collection's set is carried over. Resolution of such conflicts may require user intervention via a supplementary dialog (Figure 3). Any choices made are remembered for subsequent file copies.

The *Enrich* panel allows the metadata set in use for the collection to be edited. For example, new elements can be added, or new values can be added to the set of existing values for an element. If the element's values have a hierarchical structure, the hierarchy can be edited using the same editor that is used when editing the metadata set.

Operations at this stage include:

- Assigning new and existing metadata values to documents.
- Assigning metadata to an individual document or to a directory of documents (including nested documents).
- Assigning hierarchical metadata, whose structure can be dynamically updated if required.
- Editing or updating assigned metadata.
- Reviewing the metadata assigned to a selection of files and directories.

For our walkthrough example, in Figure 5 the user has selected the document *ac01ne.htm* and assigned "The Courier" as its *Magazine* metadata. The buttons for updating and removing metadata become active depending on what selections have been made.

During the enrichment phase, or indeed at any other time, the user can choose to view all the metadata that has been assigned to documents in the collection. This is done by selecting *Metadata* from the main menu bar,

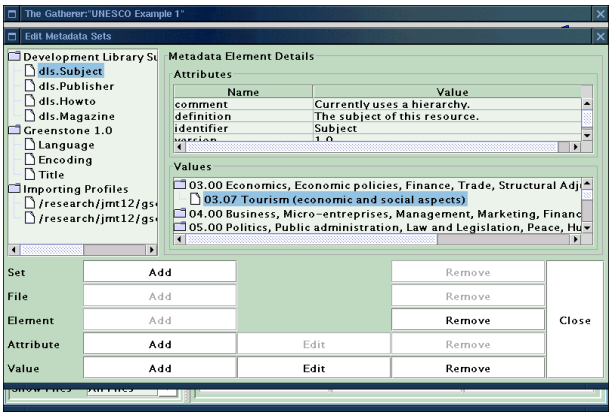


Figure 7. Editing the metadata set

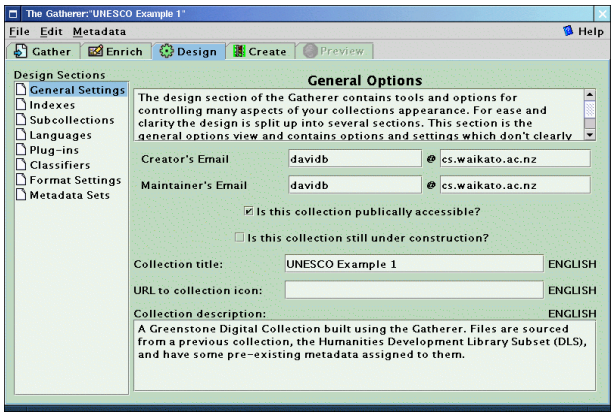


Figure 8. Designing the collection

which brings up a popup window like that in Figure 6 that shows the metadata in spreadsheet form. For large collections it is useful to be able to view the metadata associated with certain document types only, and if the user has specified a file filter as mentioned above, only the selected documents are shown in the metadata display.

The panel in Figure 7 allows the user to edit metadata sets. Here, the user is looking at the *Subject* element of the DLS set. The values of this element form a hierarchy, and the user is examining, and perhaps changing, the list of values assigned to it. The same panel also allows you to change the "profile" for mapping elements of one metadata set to another. This profile is created when importing documents from collections that have pre-assigned metadata.

## 1.3 Designing the collection

The *Design* panel (Figures 9–10) allows one to specify the structure, organization, and presentation of the collection being created. As noted earlier, the result of this process is recorded in a "collection configuration file," which is Greenstone's way of expressing the facilities that a collection requires. This step involves a series of separate interaction screens, each dealing with one aspect of the collection design. In effect, it serves as a graphical equivalent to the usual process of editing the configuration file manually.

Operations include:

- Reviewing and editing collection-level metadata such as title, author and public availability of the collection.
- Defining what full-text indexes are to be built.
- Creating sub-collections and having indexes built for them.
- Adding or removing support for predefined interface languages.
- Constructing a list of plug-ins to be used, and their arguments.
- Presenting the list to the user for review and modification.
- Configuring individual plug-ins.
- Constructing a list of "classifiers," their arguments, assignment and configuration.
- Assigning formatting strings to various controls within the collection, thus altering its appearance.
- Reviewing the metadata sets, and their elements, used in the collection.

In Figure 8 the user has clicked the *Design* tab and is reviewing the general information about the collection, entered when the new collection was created. On the left are listed the various facets that the user can configure: Indexes, Subcollections, Languages, Plug-ins, Classifiers, Format Settings, and Metadata Sets. Appearance and functionality varies between these. For example, clicking the *Plug-in* button brings up the screen shown in Figure 9, which allows you to add, remove or configure plug-ins, and change the order in which the plug-ins are applied to documents.

Both plug-ins and classifiers have many different arguments or "options" that the user can supply. The dialog box in Figure 10 shows the user specifying arguments to some of the plug-ins. The grayed-out fields become active when the user adds the option by clicking the tick-box beside it. Because Greenstone is a continually growing open-source system, the number of options tends to increase as developers add new facilities. To help cope with this, Greenstone has a "plug-in information" utility program that lists the options available for each plug-in, and the Gatherer automatically invokes this to determine what options to show. This allows the interactive user interface to automatically keep pace with developments in the software.

## 1.4 Building the collection

The *Build* panel (Figure 11) is to construct a collection based on the documents and assigned metadata. The brunt of this work is borne by the Greenstone code itself (which must be installed for the Gatherer to perform this step). The user controls this

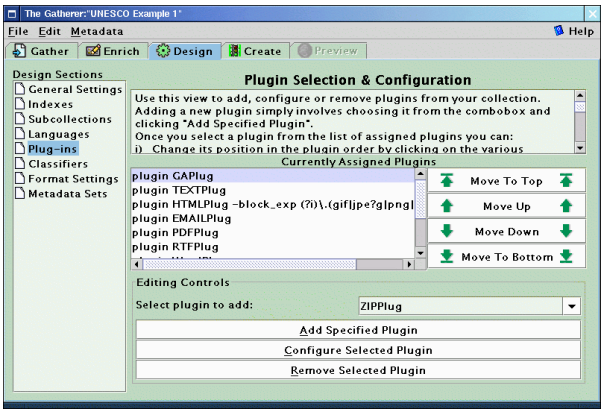


Figure 9. Specifying which plug-ins to use

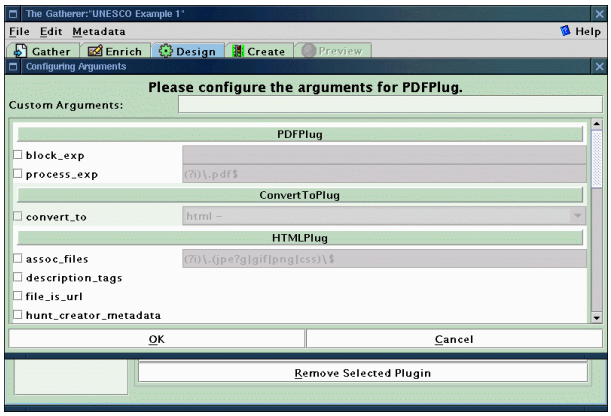


Figure 10. Configuring the arguments to a plug-in

external process through a series of separate interaction screens, each dealing with the arguments provided to a certain stage of the creation process.

The user observes the building process though a window that shows not only the text output generated by Greenstone's importing and index-building scripts, but also progress bars that indicate the overall degree of completion of each script.

Figure 11 shows the *Create* view. On the left are shown groups of options that can be applied during the creation process: General, Import, Build, All, Message-Log. The user selects appropriate values for the options.

This figure illustrates a popup "tool tip" that is available throughout the Gatherer to explain the function of each argument. In this case the user is selecting the verbosity level for the collection creation process, and the tool tip advises that this is a number between zero and three that controls how much information about the process is printed to the standard error stream—0 gives a little, 3 gives lots.

When satisfied with the arguments, the user clicks *Build Collection*. Greenstone programs continually print text that indicates progress, and the Gatherer shows these, along with a more informative progress bar.

### 1.5 Previewing

The *Preview* panel (Figure 12) is to view the collection that has been built. The Gatherer provides a preview in a panel that looks just like a web browser. In practice, previewing often shows up deficiencies in the collection design, or in the individual metadata values, and the user frequently returns to earlier stages to correct these. This tab becomes active once the collection has been created.

### 1.6 Help

The Gatherer incorporates extensive on-line help, which is invoked using the *Help* item at the right of the main menu bar at the top of each of the Figures. This opens up a hierarchically structured file of help text, and account is taken of the user's current context to highlight the section that is appropriate to the present stage of the interaction. Furthermore, as noted above, whenever the mouse is held still over any interactive object a small window pops up to give a textual "tool tip," as illustrated in Figure 11.

## 1. Configurability

Each system reviewed under "Related work" in Section 3 adopts a particular style of workflow, a natural consequence of the focused application domain that motivated the project. However, our requirements include the more ambitious goal of supporting multiple workflow styles. This is accomplished through a "preferences" screen that allows users to choose which panels appear. Panels can be activated or deactivated according to the desired work flow, which—in combination with importing and exporting options—adjusts the entry and exit points of the interactive process. Here are some examples.

Suppose the Gatherer is configured to show the first two panels (*Gather* and *Enrich*) at remote sites and the last four (*Enrich*, *Design*, *Create*, *Preview*) at a central server (where Greenstone is also installed), and data exchange is activated using the file-export menu option. This gives the same workflow as that used by Mercy Corps' *Metadata Creator* (Section 3.4). The metadata (*Enrich*) panel occurs in both the remote and central configurations, permitting remote users to enter metadata and a central librarian to make editorial corrections. For maximum flexibility the remote site version could be packaged an applet delivered from the central site that used a certificate to request permission to access the remote user's local file system. EPrints' submission process can be accomplished using the same mechanism. Since the Gatherer supports different metadata sets, it could be used in both of these very different application contexts.

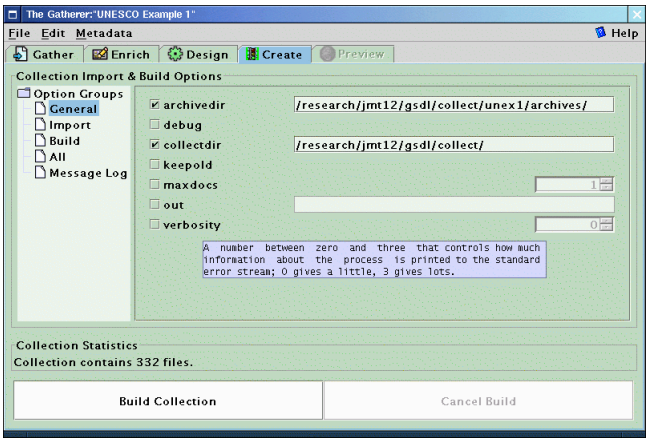


Figure 11. Getting ready to create the new collection

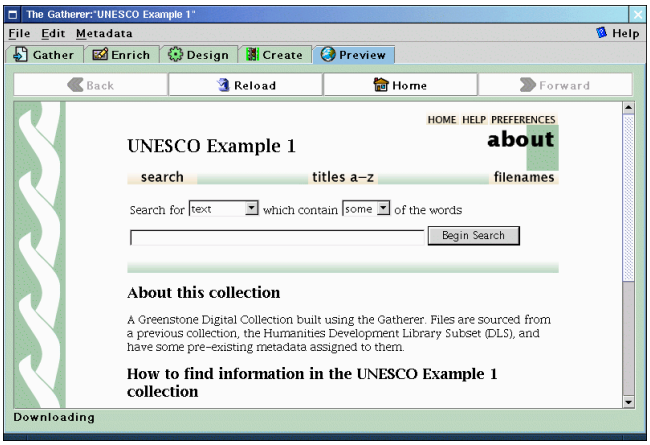


Figure 12. Previewing the newly built collection

Configuring the Gatherer to include web mirroring support provides comparable abilities to Harvest. Two new panels appear when this option is switched on. The first is used to browse the web. Once a page, set of pages or site of interest has been located, the "mirror" panel is used to control what is downloaded. This naturally flows onto the *Gather* panel, already discussed, where material is selected for inclusion in a collection.

Dispensing with the *Gather* and *Enrich* panels allows catalog-only datasets (produced using, for example, CDS/ISIS) to be incorporated by specifying the appropriate plug-in in the *Design* panel. Alternatively these panels can be switched back in and the file-import menu used instead. This allows individual records within the catalog to be viewed and edited in the *Enrich* panel.

Providing the functionality of a regular library catalog system like Koha is more challenging because additional infrastructure needs to be included in Greenstone. Despite the very different context, this could be accomplished using Greenstone's extensible system of plug-ins and classifiers. Custom plug-ins would be required to manipulate the relevant kind of datasets. Each Koha database would be viewed as a Greenstone "collection," and each item in it as a document with metadata.

For example, library membership information could be accessed by building a collection whose documents are personal records with metadata that records their name, membership number, and books borrowed. Then the Gatherer can be used to operate and maintain the system. Data is entered and edited through the *Enrich* panel, and users change collection to switch to a different database.

## 1. Conclusion

The goal of the Gatherer is to provide an interactive environment for collection editors who build digital library collections using Greenstone. We have identified the requirements for such a system. Prominent amongst these are the need to be able to recruit documents into the collection from a variety of sources; the ability to assign metadata sets, modify them if necessary, and assign metadata to documents interactively; and the process of designing, configuring, building, and testing the collection itself. Other requirements include

- creating and editing metadata
- being able to work with different metadata sets
- capability of assimilating new features of Greenstone (plug-ins and classifiers) without the need for specific program modifications.

The Gatherer has been designed to satisfy these requirements. It guides the user step by step through the process of defining and building a digital library collection. Although formal user studies have yet to be undertaken, informal user trials indicate that it provides an environment for collection-building that is superior to anything previously available. We are currently in the process of establishing field trials with several international users. The name "Gatherer" was coined to describe a much earlier version of the design and is somewhat inappropriate now: ultimately this will probably become known simply as the librarian's interface to Greenstone.

Although the Gatherer is tightly integrated into the Greenstone system, it is possible to run the early stages of document gathering and metadata editing without Greenstone being installed on the user's computer. This allows many different workflow configurations (for example, remote users, central librarian) and also raises the possibility of exporting documents and metadata for other digital library systems. One of the strengths of the Gatherer is its ability to use "Preferences" selections to emulate workflow models built into other systems.

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.

The Gatherer represents a significant enhancement to Greenstone. Although it is specific to this particular digital library software, we hope that the ideas show how such an interface can be applied more widely to provide interactive collection-building facilities in other digital library systems.

## Acknowledgements

Greenstone is the work of a great many people and we gratefully acknowledge the contributions of them all, and in particular the stimulating research environment in the Digital Library Lab at Waikato. We are also grateful to the anonymous reviewers for helping us to focus our message.

## 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] Blake, R. and Hamilton-Williams, R. (2000) "The Koha story." http://koha.org/about/story.html.

[3] Buxton, A. and Hopkinson, A. (2001) *The CDS/ISIS for Windows Handbook*. UNESCO, Paris, September.

[4] Cole, H., Eckstein, R., Elliott, J., Loy, M. and Wood, D. (2002) *Java Swing*. O'Reilly and Associates.

[5] Consultative Committee for Space Data Systems (CCSDS) (2002) *Reference model for an Open Archival Information System (OAIS)*. CCSDS 650.0-B-1 Blue Book, Washington, DC.

[6] Crane, G. and Rydberg-Cox, J.A. (2000) "New technology and new roles: the need for 'corpus editors'." *Proc Digital Libraries 2000*, San Antonio, Texas, pp. 252–253.

[7] Hardy, D.R., Schwartz, M.F. and Wessels, D. (1996) Harvest 1.4 User's Manual. Technical Report CU-CS-743-94, University of Colorado. Revised by K.-J. Lee in 2002 for Harvest version 1.8.

[8] Harnad, S. & Carr, L. (2000) "Integrating, navigating, and analysing open Eprint archives through open citation linking (the OpCit project)" *Current Science*, Vol. 79, No.5, pp. 629-638.

[9] Hitchcock, S., Carr, L., Jiao, Z., Hall, W. and Harnad, S. "Developing Services for Open E-Print Archives: Globalisation, Integration and the Impact of Links." *Proc. ACM Digital Libraries Conference*, San Antonio, Texas, 2000, ACM Press: New York.

[10] Kenney, A.R. and Rieger, O.Y. (2000) *Moving theory into practice: digital imaging for libraries and archives*. Research Libraries Group, Mountain View, CA.

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

[12] Loots, M., Carmazan, D. and Witten, I.H. (2002) "From paper to collection." Greenstone Digital Library software manual, available at greenstone.org.

[13] Witten, I.H., Loots, M., Trujillo, M.F. and Bainbridge, D. (2002) "The promise of digital libraries in developing countries." *The Electronic Library*, Vol. 20, No. 1, pp. 7-13.

[14] 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.

[15] Witten, I.H., Bainbridge, D. and Boddie, S.J. (2001) "Power to the people: end-user building of digital library collections." *Proc Joint Conference on Digital Libraries*, Roanoke, VA, 94-103; June.

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

[17] Wolpert, A.J. (2002) "The future of electronic data." *Nature* Vol. 420, pp. 17-18.