APPLICATION OF DCMI IN OPEN SOURCE SOFTWARE WITH SPECIALS REFERENCE TO GSDL: A CRITICAL STUDY

Subarna Kumar Das Biswajit Das Rajesh Das Subhendu Kar

Abstract

This paper is discussed about how to incorporate 15 Dublin Core Meta data element set in Greenstone digital library software (open source) for more robust flexible search in an Institutional Repository. Here, we have created a framework for integration GSDL Meta data with Dublin Core meta data using a Meta data creator tool software named DC-dot. This tool helps to convert Dublin Core Metadata to Greenstone XML documents and built into a collection.

Keywords : Digital Library, Institutional Repository, Open Source Softwares

1. INTRODUCTION

This is very essential for creating and delivering digital collections; we need robust and flexible digital collections management and presentation software. But digital library technologies and contents are not static. Continual evolution and investment are required to maintain the digital library. Few commercial digital library products are comprehensive and extensible enough to support this evolution. Open source applications in particular allow developers and users to modify the system and tailor it to their own particular needs. Like commercial software, open source software will not be a perfect solution. But open systems at least give developers and users the opportunity to modify functionality and create interfaces for integration with other software. Open source digital library software allows incorporating the Meta data set according to DCMI or any others for flexible information retrieval. The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems. Other hand, Greenstone digital library software is developed by the New Zealand Digital Library Project at the University of Waikato. It has many good features that meet our requirements, including a powerful search engine (mg) and metadata-based browsing facilities. But it lacks a good metadata management interface based on the Dublin Core standard, so we customized Greenstone to use the Dublin Core metadata.

2. DUBLIN CORE META DATA INIACITIVE (DCMI)

The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems.

The mission of DCMI is to make it easier to find resources using the Internet through the following activities:

1. Developing metadata standards for discovery across domains,
2. Defining frameworks for the interoperation of metadata sets, and,
3. Facilitating the development of community- or disciplinary-specific metadata sets that are consistent with items 1 and 2

The range of activities of DCMI includes:

- Standards development and maintenance, such as organizing international workshops and working group meetings directed toward developing and maintaining DCMI recommendations.
- Tools, services, and infrastructure, including the DCMI metadata registry to support the management and maintenance of DCMI metadata in multiple languages.
- Educational outreach and community liaison, including developing and distributing educational and training resources, consulting, and coordinating activities within and between other metadata communities.

3. DUBLIN CORE META DATA SET ELEMENT

The Dublin Core metadata element set is a standard for cross-domain information resource description. Here an information resource is defined to be “anything that has identity”. This is the definition used in Internet RFC 2396, “Uniform Resource Identifiers (URI): Generic Syntax”, by Tim Berners-Lee et al. There are no fundamental restrictions to the types of resources to which Dublin Core metadata can be assigned.

The Elements:

i) Element Name: *Title*

Label: Title

Definition: A name given to the resource.

Comment: Typically, Title will be a name by which the resource is formally known.

ii) Element Name: *Creator*

Label: Creator

Definition: An entity primarily responsible for making the content of the resource.

Comment: Examples of Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.

iii) Element Name: *Subject*

Label: Subject and Keywords

Definition: A topic of the content of the resource.

Comment: Typically, Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.

iv) Element Name: *Description*

Label: Description

Definition: An account of the content of the resource.

Comment: Examples of Description include, but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.

v) Element Name: *Publisher*

Label: Publisher

Definition: An entity responsible for making the resource available

Comment: Examples of Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.

vi) Element Name: *Contributor*

Label: Contributor

Definition: An entity responsible for making contributions to the content of the resource.

Comment: Examples of Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.

vii) Element Name: *Date*

Label: Date

Definition: A date of an event in the lifecycle of the resource.

Comment: Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and includes (among others) dates of the form YYYY-MM-DD.

viii) Element Name: *Type*

Label: Resource Type

Definition: The nature or genre of the content of the resource.

Comment: Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DCMI Type Vocabulary [DCT1]). To describe the physical or digital manifestation of the resource, use the FORMAT element.

ix) Element Name: *Format*

Label: Format

Definition: The physical or digital manifestation of the resource.

Comment: Typically, Format may include the media-type or dimensions of the resource. Format may be used to identify the software, hardware, or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [MIME] defining computer media formats).

x) Element Name*: Identifier*

Label: Resource Identifier

Definition: An unambiguous reference to the resource within a given context.

Comment: Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Formal identification systems include but are not limited to the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).

xi) Element Name*: Source*

Label: Source

Definition: A Reference to a resource from which the present resource is derived.

Comment: The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.

xii) Element Name: *Language*

Label: Language

Definition: A language of the intellectual content of the resource.

Comment: Recommended best practice is to use RFC 3066 [RFC3066] which, in conjunction with ISO639 [ISO639]), defines two- and three-letter primary language tags with optional subtags. Examples include “en” or “eng” for English, “akk” for Akkadian”, and “en-GB” for English used in the United Kingdom.

xiii) Element Name: *Relation*

Label: Relation

Definition: A reference to a related resource.

Comment: Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.

xiv) Element Name: *Coverage*

Label: Coverage

Definition: The extent or scope of the content of the resource.

Comment: Typically, Coverage will include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN]) and to use, where appropriate, named places or time periods in preference to numeric identifiers such as sets of coordinates or date ranges.

xv) Element Name: *Rights*

Label: Rights Management

Definition: Information about rights held in and over the resource.

Comment: Typically, Rights will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions may be made about any rights held in or over the resource.

4. CUSTOMIZATION DUBLIN CORE META DATA AND GREENSTONE DIGITAL LIBRARY

C1. DC-dot (A Dublin Core meta data creation tool)

DC-dot is a tool software for Web-based Dublin Core generating and editing, developed by Andy Powell at UKOLN, University of Bath, United Kingdom. A user can enter a Web page URL and DC-dot then captures information from the Web page and generates Dublin Core metadata automatically. The metadata is presented to the user in a Web form for manual enhancement. Here we adopted the Dublin Core data

entry form, added several features, integrated it with Greenstone's collection management tools, and are using it for our metadata creation and management interface.

DC-dot was not built to be extensible, so we could not avoid some changes to its CGI Perl script, dcdot.pl. This tool software was designed to describe HTML pages by default, but we have described for other digital objects such as image files. So we modified dcdot.pl to recognize a new kind of metadata file (identified by the .dc extension). For each object to describe, a metadata file is created from a template with certain fields pre-populated with standard values for that collection. DC-dot reads the metadata file and presents the Web form for additional data entry. We modified DC-dot to look for files in our image repository and, if found, add a link to the form for the metadata entry staff to use to view the image being described. With these relatively few modifications we were able to use DC-dot to enter and maintain metadata for our digital collections.

A serious limitation of DC-dot was that the unqualified Dublin Core metadata it generates is not rich enough to describe the detail we wanted for our collections. An important enhancement was to add arbitrary qualifiers to Dublin Core fields. To minimize changes to the dcdot.pl script, we developed a separate Perl module to "override" some of the DC-dot functions (particularly the ones that read and write the metadata) so they could recognize and handle Dublin Core qualifiers. When processing a .dc file, dcdot.pl will call the module routines for these functions instead of the local ones. We also provided a new function to write the HTML for the DC-dot data entry form. Besides handling qualifiers, this routine builds drop-down pick lists from authority files.

Other enhancements to the metadata creation and maintenance component are provided by a set of CGI Perl scripts that manage the Dublin Core records. Our metadata repository consists of files organized in separate file system directories for each collection. Each metadata file represents a Dublin Core record. dcnew.pl generates a new metadata file from a template. This script can be used to create a meta-record (one that describes other records rather than a digital object) or to create a new template. dcobj.pl lists objects that haven't yet been described and generates a new metadata file for a selected object. It also scans existing records to rebuild authority files to populate the drop-down lists for data entry. dcupd.pl lists objects that have been described, so a selected record can be updated. dcsrch.pl provides a simple search mechanism to help locate a record to be updated. All these scripts provide links to dcdot.pl to display a Dublin Core record for data entry and update. The relationships between these scripts are shown in Figure 1.

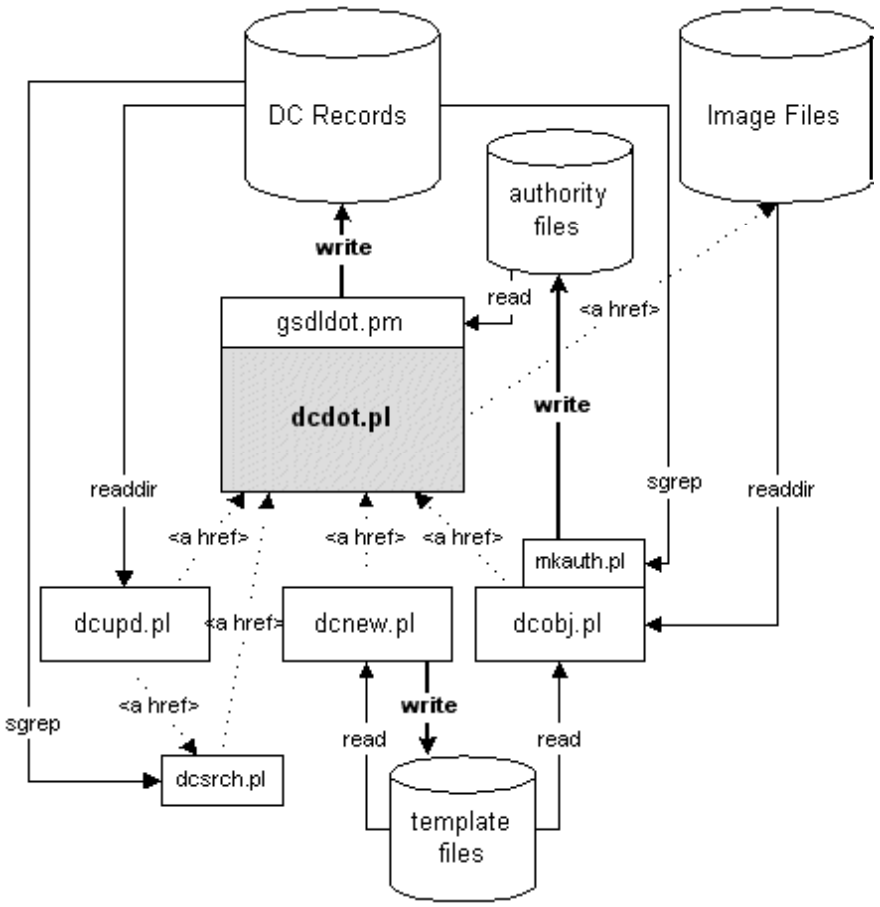

Figure 1: DC-dot and auxiliary scripts for creating and editing metadata

Other scripts were created to manage the process of importing the Dublin Core records into Greenstone collections. A CGI script allows an administrator to update the Greenstone configuration file, import a set of Dublin Core records, and build a collection. The configuration file is difficult to edit because line breaks and indentation are not allowed in directives. In particular, the format directives can be quite complex requiring very long lines that are difficult to understand and maintain. The CGI script presents each format directive in a separate Web form text box with line breaks and indentation. It uses make_cfg.pl to strip new line characters and merge them into the configuration file.

Before importing Dublin Core records into Greenstone, the administrative script preprocesses them to enhance the structural metadata. Each record has structural metadata fields to specify parent and child relationships. During metadata creation, a child's parent can be specified in its *Relation.parent* field.

Rather than require that the same information be entered in the parent record for each child, all the children of a parent are identified during import preprocessing and added to the parent's *Relation.children* field. This allows links to be created in Greenstone in both directions (up and down the hierarchy).

C2. Greenstone Digital Library

Greenstone digital library software is developed by the New Zealand Digital Library Project at the University of Waikato. It has many good features that meet our requirements, including a powerful search engine (mg) and metadata-based browsing facilities. But it lacks a good metadata management interface based on the Dublin Core standard, so we customized Greenstone to use the Dublin Core metadata from DC-dot.

In contrast to DC-dot, Greenstone was designed to be highly extensible and to handle arbitrary kinds of metadata. A variety of plugins are available to parse input documents and extract metadata from them. Custom plugins can be written to extract different kinds of metadata. Perl object-oriented features allow new plugins to inherit from existing ones, as we did for a Dublin Core plugin that inherits from their HTMLPlug.pm. DCPlug.pm overrides the object constructor and the process() and extract_metadata() methods to parse and process the Dublin Core metadata produced by DC-dot. Figure 2 shows how the Dublin Core plugin works with the Greenstone collection import and build programs, under the control of our administrator CGI script.

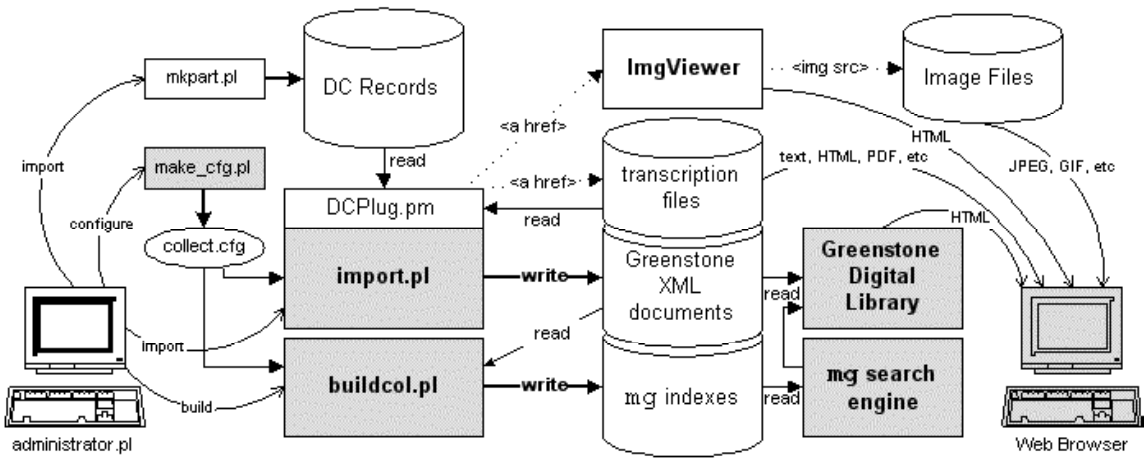

Figure 2: Greenstone and auxiliary scripts for building and presenting digital collections

DCPlug.pm automatically enhances the metadata to create links to digital objects and to overcome some limitations in the Greenstone search engine. Greenstone is limited in how it can handle repeating fields and search across multiple fields. DCPlug.pm has an option to specify fields whose values should be accumulated into new fields that Greenstone can display. Another option allows multiple fields to be accumulated into a new field that can be indexed for a keyword search. As with other Greenstone plugins, the DCPlug.pm options are specified in the collection configuration file.

Comprehensive Digital Collections Management System

Integration and customization of the open source software systems was more difficult than we wished or expected. But the result of our efforts is a fully functional, flexible and powerful digital collections management system that is tailored to our local environment and organizational needs. The system consists of a metadata creation tool, an administration tool and an attractive Web interface.

The features of the metadata creation tool include:

- **Digital object identification:** A list of digital objects available for cataloging is automatically generated with a link to view the object. Once the object is cataloged, the object and related images are removed from this list.
- **Local authority control:** The data entry form includes drop-down pick lists for selecting standard metadata values, such as personal names, subjects, material types, and so on. Authority files can be created from standard vocabularies, or automatically generated from existing metadata in the collection. Data entry staff can pick authority values from the list resulting in simpler data entry and fewer errors.
- **Metadata editing:** DC-dot and auxiliary tools allow searching, jumping to a record, editing and deleting records, and adding new fields.
- **Template creation:** Templates for various kinds of records can be created and used to automatically generate metadata for Dublin Core fields.
- **Digital object access:** Master and derivative image files can be viewed and retrieved.

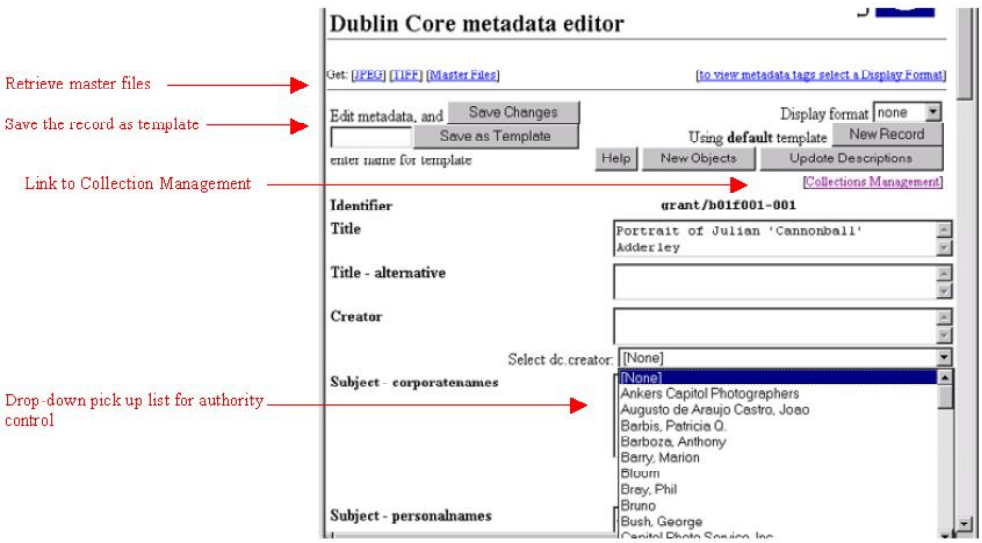

Figure 4: shows some of these features

The features of the administration tool include:

- **Collection configuration:** A Web form simplifies the creation and editing of the Greenstone configuration file.
- **Importing Dublin Core records:** Metadata from DC-dot are converted to Greenstone XML documents and built into a collection.
- **Global changes:** A Perl script can be used to change or delete the value of a metadata field for all the records in a collection.
- **Nightly rebuilds:** An automatic import and build tool processes the collections that have been updated each day by the owning library staff.

The Greenstone user interface was customized to highlight the unique features of the individual digital collections. The metadata description is presented in a standard library OPAC format with a thumbnail image. The full-size images in the digital object can be viewed with Image Viewer in another browser window. Full-text transcriptions in any formats are linked within the record and can be viewed through appropriate applications.

5. CONCLUSION

As we develop more complex and large digital collections, we are finding that the file system-based repository for our digital objects and metadata is getting more difficult to manage. We are now investigating the addition of a database or XML driven repository. It is testing Fedora. a repository for digital objects based on the METS encoding scheme. METS would allow us to encapsulate all the metadata for a digital object in a single standard package without the (sometimes) awkward qualifiers used to encode it in Dublin Core. We would keep our descriptive metadata in Dublin Core while using more appropriate schemes for structural, administrative and behavioral information. This would also allow us to easily implement additional interfaces to the metadata, so our digital objects can be part of larger virtual and distributed collections.

6. REFERENCES

1. Aas, Gisle, HTML::TokeParser, http://search.cpan.org/author/GAAS/HTML-Parser-3.26/ (Search on 11-09-2005)
2. Barkstrom, Bruce, M. Finch, M. Ferebee, & C. Mackey, Adapting Digital Libraries to Continual Evolution, in Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries, July 14-18, 2002, Portland Oregon.
3. California Digital Library, Online Archive of California Customized Templates for Finding Aids, http://www.oac.cdlib.org/templates/uctemp.html (Search on 11-09-2005)
4. Dublin Core Metadata Initiative, http://dublincore.org/ (Search on 11-09-2005)
5. Encoded Archival Description, http://lcweb.loc.gov/ead/ (Search on 11-09-2005)
6. Gourley, Don, An Architecture for the Evolving Digital Library, in EDUCAUSE Information Resources Library, November 26, 2001, http://www.educause.edu/ir/library/html/edu0122/edu0122.html (Search on 11-09-2005)
7. Greenstone Digital Library Software, http://www.greenstone.org/ (Search on 11-09-2005)
8. Hulse, Bruce, & Elizabeth Payne, Proposal to Institute of Museum Library Services 2001 National Leadership Grants for Libraries, http://www.wrlc.org/dcpc/imlsproposal.pdf (Search on 11-09-2005)

9. Jaakkola, Jani & Pekka Kilpeläinen, sgrep, http://www.cs.helsinki.fi/u/jjaakkol/sgrep.html (Search on 11-09-2005)

10. Lund, William, Digital Object Library Products, in *RLG DigiNews*, October 15 2001, http://www.rlg.org/preserv/diginews/diginews5-5.html (Search on 11-09-2005)

11. Metadata Encoding and Transmission Standard, http://www.loc.gov/standards/mets/ (Search on 11-09-2005)

12. The Open Archives Initiative, http://www.openarchives.org/(Search on 11-09-2005)

13. Paynter, Gordon, Making Complex Formatstrings Slightly Less Complex, in Greenstone User List Archive, http://www.nzdl.org/cgi-bin/library?a=p&p=about&c=gsarch (Search on 11-09-2005)

14. The Perl Foundation, http://www.perl-foundation.org/ (Search on 11-09-2005)

15. Powell, Andy, DC-dot, http://www.ukoln.ac.uk/metadata/dcdot/ (Search on 11-09-2005)

16. University of Virginia & Cornell University, Fedora, http://www.fedora.info/ (Search on 11-09-2005)

About Authors

Subarna Kumar Das is a Senior Lecturer of Department of Library & Information Science, Jadavpur University, Kolkata , West Bengal.

Biswajit Das is a Student of PGDDLM of Department of Library & Information Science, Jadavpur University, Kolkata , West Bengal.

Rajesh Das is a Student of PGDDLM of Department of Library and Information Science, Jadavpur University, Kolkata , West Bengal.

Subhendu Kar is a Student of PGDDLM of Department of Library & Information Science, Jadavpur University, Kolkata , West Bengal.