Using Dublin Core to Facilitate Cross-Collection Searches in an Enterprise Image Repository

Kevin J. Hegg, Andreas R. Knab, Center for Instructional Technology, James Madison University, USA
{heggkj, knab2ar}@jmu.edu

Abstract

The Madison Digital Image Database (MDID) is an Internet-based content delivery system designed to allow instructors to teach with digital images and image catalog data. James Madison University (JMU) is currently developing an enterprise-level version of the MDID that will support custom catalogs, multiple collections, and cross-collection searches. The new MDID relies on Dublin Core (DC) to facilitate cross-collection searches. We anticipate that Dublin Core will also make catalog customization, data exchange and system interoperability easier. This poster describes the implementation of Dublin Core in a relational database, the mechanism for conducting cross-collection searches, and other benefits of using Dublin Core in the new MDID.

Keywords: *Relational Database, Collection, Image, Search, Madison Digital Image Database*

1. Introduction

JMU developed the MDID in 1998 in response to increased enrollment in Survey of World Art courses [1]. The software was released to the public, free of charge, in October 2001. Today, more than two dozen institutions are actively using it.

The MDID is comprised of a web application and a standalone application—the "ImageViewer." Instructors use the web application to search for images and build slideshows. Images within a slideshow are then sorted and annotated. Students use the web application to review and study slideshows. The ImageViewer is used in the classroom to retrieve slideshows over an intranet and to display images and catalog data at full screen resolution with an LCD projector.

Since its introduction, the MDID has undergone a few minor changes; however, its underlying technology remains unchanged. The MDID in its current state supports only one image collection with a fixed catalog structure. In an effort to update its technology and make it more flexible, we began designing and developing a new version of the MDID in the summer of 2002. With the support of a grant from the Andrew W. Mellon Foundation, we were able to accelerate development in January 2003.

The new MDID will support multiple collections, each with a custom catalog. We anticipate releasing the new MDID to the public under an open source license in the fall of 2003.

2. Dublin Core in a Relational Database

One of the basic principles of Dublin Core states that "each element is optional and repeatable" [2]. Relational databases do not support multi-valued or repeated columns. There are several ways to circumvent this restriction. For example, multiple values can be concatenated within a single column using a delimiter. Alternatively, multiple values for the same column can be stored in a separate table and referenced from the catalog table. However, both of these solutions oblige the programmer to build complex and unintuitive queries.

We took another approach to storing repeatable elements in a relational database. All values are placed in a *field values* table.

Table 1. Data Structure of Field Values Table

Column Name	Column Description
Image	Identifies the visual resource
Field	Refers to a catalog field
Instance	Identifies a single instance
Value	Contains a single field value

Each record in the *field values* table represents an individual value. The record contains references to the relevant image and field. It also holds an instance identifier that distinguishes between multiple values and preserves their order. This approach stores each value in its own uniquely identifiable row and allows us to build straight-forward queries. Keeping the entire catalog data in one table also limits any cross-collection, full-text keyword search to one table, making queries easier to construct and more efficient.

3. Multiple and Custom Catalogs

In addition to the *field values* table, the MDID uses a table that defines the fields in every catalog—the *field definition* table (see table 2). Each record in this table represents a single field in a catalog. The value of the Name column is curator-defined and is not restricted in any way. The DC Element and DC Refinement columns are optionally used to map the field to a Dublin Core element or refinement [3]. Each DC element or refinement can only be mapped to one catalog field. Not all fields have to be mapped to a DC element or refinement.

https://doi.org/10.23106/dcmi.952107515

Table 2. Data Structure of Field Definitions Table

Column Name	Column Description
Catalog	Identifies the catalog
Name	Field name
DC Element	Corresponding DC element
DC Refinement	Corresponding DC refinement
Order	Defines field order

Together, the *field values* and *field definitions* tables allow us to define and store the content of any number of curator-defined catalogs without having to create additional tables for new catalogs. This approach eliminates the need to change the database structure at any time.

4. Cross-Collection Searches

While users will often conduct searches within a single collection, the new MDID provides a powerful mechanism to extend searches across related but distinct collections.

To perform a cross-collection search, the user first selects two or more collections. The system then determines which fields are common to all selected collections by comparing the DC Element and DC Refinement values. If two fields from two different collections are mapped to the same element or same element-refinement pair, their contents are assumed to be comparable. The user is presented with a search form containing the common fields. After the user enters search terms, the search is run against the matching fields in the *field values* table.

The process of determining common fields across multiple collections using Dublin Core mappings is illustrated in figure 1.

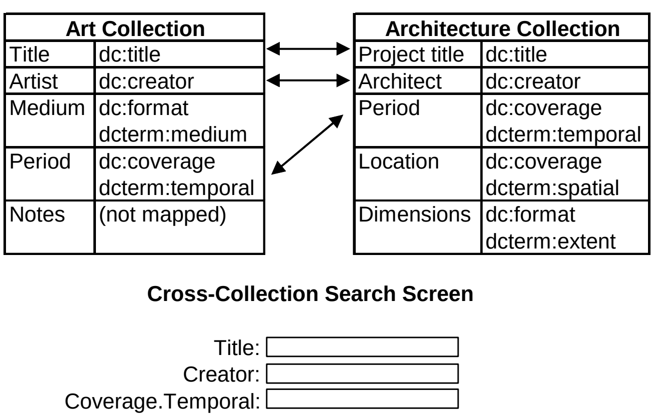

Figure 1. Using Dublin Core to Determine Common Fields

We use Dublin Core labels on the Cross-Collection Search Screen because curator-defined field labels are not necessarily identical across collections.

Because curators map fields to DC elements and refinements independently of one another, semantic relationships between the collections form naturally and without the intervention of a "super-curator."

5. Data Exchange

Mapping curator-defined fields to DC elements and refinements is also valuable in exchanging data between the MDID and external systems. When importing data that validates against the Dublin Core XML schema [4], the MDID will automatically map the DC elements to the appropriate MDID fields. Conversely, when exporting data for use in a Dublin Core-compliant system, the mapped DC element and refinement names will be written in place of the curator-defined field names. The resulting XML will validate against the Dublin Core XML schema.

Future plans for the MDID include the addition of mechanisms to support interoperability between the MDID and remote databases. We envision employing an emerging protocol such as ZING SRW [5] to facilitate interoperability. All SRW-compliant databases must support the Simple Dublin Core schema [6]. We are able to meet this requirement because curator-defined fields are mapped to DC elements. If an external database supports SRW, MDID users will be able to search its catalogs remotely and users of the external database—for example, another MDID installation—will be able to search MDID catalogs.

6. Conclusion

The Dublin Core offers an elegant solution to the problem of finding common semantic ground between different collections. We are able to provide a more precise means of searching across collections than performing simple keyword searches. Of course, this solution relies on the Dublin Core's ability to describe commonalities across disciplines and on the curator's ability to accurately map the MDID fields to DC elements and refinements.

References

[1] *The Madison Digital Image Database (MDID).* (n.d.). Retrieved May 14, 2003 from http://cit.jmu.edu/mdidinfo/.

[2] Hillmann, D. I. (2001). *Using Dublin Core.* Retrieved April 28, 2003, from Dublin Core Metadata Initiative Web site: http://www.dublincore.org/documents/2001/04/12/usageguide/.

[3] DCMI Usage Board. (2003). *DCMI Metadata Terms.* Retrieved May 13, 2003, from Dublin Core Metadata Initiative Web site: http://www.dublincore.org/documents/2003/03/04/dcmi-terms/.

[4] Powell, A. & Johnston, P. (2003). *Guidelines for Implementing Dublin Core in XML.* Retrieved May 14, 2003, from Dublin Core Metadata Initiative Web site: http://www.dublincore.org/documents/dc-xml-guidelines/.

[5] *SRW/U Background – ZING (Library of Congress).* (2002). Retrieved May 14, 2003 from http://lcweb.loc.gov/z3950/agency/zing/srw/background.html.

[6] *Retrieving Records in SRW – ZING (Library of Congress).* (2002). Retrieved May 14, 2003, from http://lcweb.loc.gov/z3950/agency/zing/srw/records.html.

https://doi.org/10.23106/dcmi.952107515