

A CROSS-PLATFORM SOLUTION FOR BIBLIOGRAPHIC RECORD MANIPULATION IN DIGITAL LIBRARIES

Wendy Osborn and Daljit Kaur
Department of Mathematics
and Computer Science
University of Lethbridge
4401 University Drive West
Lethbridge, Alberta
Canada T1K 3M4
email: wendy.osborn@uleth.ca

Kathy Crewdson and Ian Dew
NextLibrary, Inc.
#273 - 1100 Memorial Avenue
Thunder Bay, Ontario
Canada P7B 4A3
email: ian@nextlib.com

ABSTRACT

We investigate the problem of a collaborative digital library collection that allows librarians and non-librarians alike to share information on specific topics through MARC records. Many existing software tools for creating bibliographic records are not platform independent. Tools that are platform independent are integrated into library software and therefore cannot be used with any existing system. J-MARC is a cross-platform tool for creating and editing bibliographic records. J-MARC can be used in conjunction with any digital library software that supports the MARC record format. We present the architecture and functionality of J-MARC, along with an example application where J-MARC can be used alongside an existing digital library software package.

KEY WORDS

digital libraries, bibliography, MARC, Greenstone, SAIR

1. Introduction

Existing software tools for creating and editing bibliographic records do not meet the needs of a wide audience, especially collaborative organizations that exist outside of a public library system. Some existing tools are integrated into existing bibliographical systems. This makes the use of such tools with other types of digital library software impossible. Other such tools are standalone applications that work on one operating system only. Since people in a collaborative environment may use computers that run different operating systems, this makes sharing information impossible. In addition, most have complex user interfaces that require knowledge of how bibliographic record formats, such as MARC, are organized.

We solve this problem with J-MARC (short for Java-MARC), a tool for creating and editing MARC records that is cross-platform, simple to use, and can be used with any digital library software system that supports the MARC format. Therefore, it allows for a community of users to be able to truly collaborate with each other. We present the architecture and functionality of J-MARC. Then, we present

an application that can use J-MARC along with its existing digital library collection to meet the needs of its wide-ranging audience of collaborators.

The remainder of this paper proceeds as follows. Section 2 contains background information on the MARC 21 bibliographic standard and the Greenstone digital library software. Section 3 presents the J-MARC bibliographic tool, including its architecture and functionality. Section 4 presents an example on how to use J-MARC to create a simple MARC file. Section 5 presents an example of the application of J-MARC in conjunction with an existing digital library software system. Section 6 presents a classification of existing MARC tools, their limitations and how J-MARC complements these tools. Finally section 7 presents conclusions and future work on J-MARC.

2. Background

In this section, we first summarize the MARC 21 bibliographic format, on which J-MARC is based. Then we summarize the Greenstone digital library software, which is an example of a system that J-MARC can complement.

2.1 The MARC 21 Standard

The Library of Congress defines Machine-Readable Cataloging (MARC) bibliographic format as a "standard for the representation and communication of bibliographic and related information in machine-readable form." [1] MARC is a standard by which many libraries organize information on the resources they maintain. This common format also allows libraries to share resource information. There exist many variations of the MARC format. One variant is MARC 21, which is widely used in North America and the United Kingdom. We will discuss the MARC 21 variant, which is the current focus of J-MARC.

A MARC 21 bibliographic record consists of several fields. A field contains the following information: a tag, indicators, subfields, and content. A tag is three digits long and serves as a numeric representation of a field label. The

two indicator values (each from 0-9, or an 'undefined' symbol) each provide further information for the field. What an indicator represents depends on the field. A field can be divided into subfields. Each subfield has a subfield code and corresponding data.

The information in a MARC 21 record is organized in the following manner. The first part of the record is called a leader. The leader is 24 characters long and contains general information about the record, such as its size and the number of fields. The next part of the record is called the tag directory. This directory contains information on each field in the record, such as its tag, the number of characters in the field, and the starting point of the content of the field. The third part of the record consists of the contents for each field listed in the tag directory. For each field, the contents consist of the indicators, the subfield codes (if any) and the data itself.

2.2 Greenstone

Greenstone [2, 3] is a software suite for creating digital library collections and making them available either locally or via the internet. An end user can create a digital library collection containing multimedia documents, such as images, PDFs, audio, and different forms of text. One type of document that Greenstone can handle is a multi-record MARC file.

Greenstone is open source, and works across different platforms. It has been tested and works on all versions of Windows (except 3.1), MAC OS X, and many flavors of Unix/Linux.

In addition, the Greenstone software and any collections built with it can be customized in many ways. For example, modules that support the inclusion of new data types can be added easily to the main software base. A collection can be customized to provide different organized views of its contents and different searching and browsing mechanisms.

One can create and edit their collection using the Greenstone Librarian Interface (GLI) [2]. For a collection of MARC records, one simply adds their MARC files to an import folder. From there, the GLI processes each files to obtain one or more MARC records, which are then added to the collection in a human readable format. By default, this is a text version of the MARC record. However, one can customize the collection to display information in other formats, such as Dublin Core [4].

3 J-MARC

In this section, we present the J-MARC software tool for creating a MARC file of records. J-MARC is designed with built-in flexibility so that some customization is possible. In addition, the flexibility allows for additional functionality to be added easily in the future. Finally, J-MARC is written in Java, so it can be used across all major platforms.

We present the underlying system structure, followed by the design of the user interface.

3.1 System Design

The underlying J-MARC system is composed of several modules. These are the File, Record, Field, Leader, Tag Directory, and Subfield. Each is described below.

- File. The J-MARC File module maintains a list of records from an existing file, or that will be placed in a new file. This list is dynamic, so new records can be added to it and will be added to the file when it is saved. The File module is responsible for creating a new file; reading and parsing an existing file; the output of all current records to either a new or existing file; and the management of the MARC record list.
- Record. The J-MARC Record module manages all aspects of a MARC record. Like the File module, the Record module maintains a flexible list of items, so new items can be added with ease. In this case, it maintains a list of MARC fields. It can be populated with fields from an existing record. The user can modify the contents of existing fields, and add new fields. For a new record, the user can select any new field to add to the field list. In addition, the Record module uses the Leader and Field Directory modules that are described below.
- Leader. The J-MARC Leader module is responsible for processing the leader.
- Tag Directory. The J-MARC Tag Directory module is responsible for processing the tag directory of a record. It identifies for each field in a record its tag, the number of characters in the field, and its location in the record.
- Field. The J-MARC Field module maintains a flexible list of subfields. It can be initially populated from an

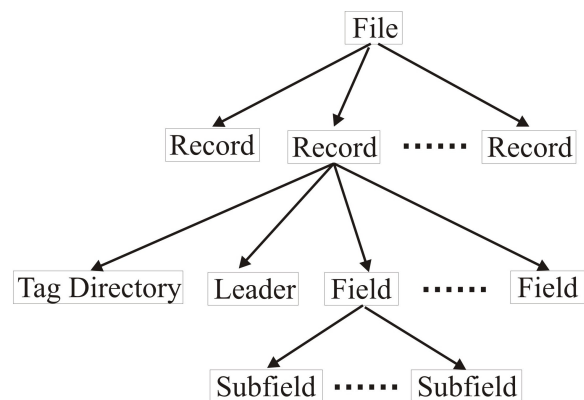


Figure 1. Organization of Modules in J-MARC

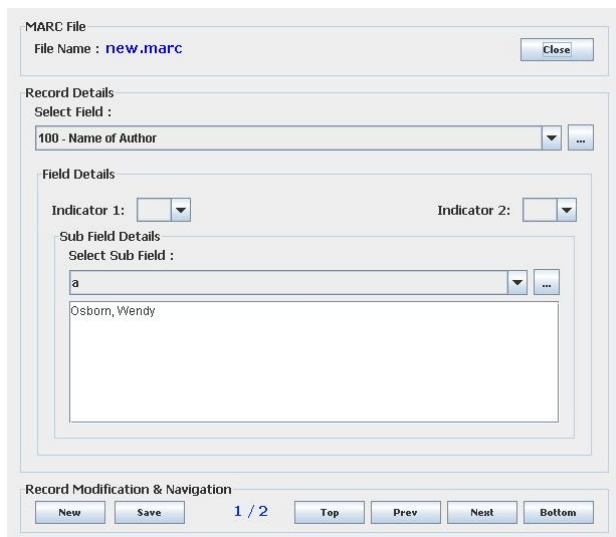


Figure 2. The J-MARC User Interface

existing field. Subfields can be added to the list in a similar manner as adding a field.

- Subfield. The J-MARC Subfield module is responsible for processing a subfield to obtain a subfield code and its corresponding contents.

Figure 1 shows how all modules are related. A File module contains one or more records, while a Record contains multiple fields (along with leader and field directory processing) and a Field module that consists of one or more subfields.

3.2 User Interface

In order for J-MARC to be useful to users who are not familiar with the MARC format, we designed a graphical user interface (GUI) that is simple to use. Only a minimal amount of information is displayed to the user at any given time. In addition, flexibility is provided in what information the user can view and modify.

Figure 2 depicts the J-MARC graphical user interface. It consists of three areas. The first area handles MARC file access. It allows the user to open either a new or existing MARC file. In both cases, the file is specified by opening a file selection dialog box and simply typing or selecting the filename. The specified file name is displayed at all times while the file is open, in addition to a button that gives the option to close and save the file.

The second area handles record details. This area allows the user to manipulate a record in the open file. Here, fields and subfields can be added, and the values of fields that exist in the record can be modified. In addition, the user can select one field and subfield to view at a time from a drop-down menu. J-MARC hides the remaining fields out of view. However, the user can switch to other fields and subfields when they are required.

Finally, the third area is for record modification and navigation. This area allows the user to create a new record, save an edited record, and navigate between all records in the file.

In addition to making the interface simple, it is also designed with flexibility with respect to the list of MARC fields that one can work with, and the labels used to identify fields. Only a small number of the many pre-defined MARC fields are used on a regular basis, while many others are used rarely [1]. Therefore, the fields that J-MARC makes available to the user are specified in a file that is read by the GUI on startup. This file contains a tag and a label for each MARC field to be included. One can use the labels as defined by the Library of Congress [1]. For example:

100-MAIN ENTRY PERSONAL NAME
 110-MAIN ENTRY CORPORATE NAME
 246-VARYING FORM OF TITLE

However, as long as the proper tags are used, the labels can be changed so that they are more meaningful to the specific group of users. For example, the fields with tags 100 and 110 can be labeled as follows:

100-Name of Author
 110-Affiliation of Author
 246-Title of Article

4. Example: Using J-MARC

In this section, we present a walkthrough that displays how J-MARC can be used to create a simple MARC file. This walkthrough involves: 1) Creating a new MARC file, 2) Opening the new MARC file, 3) Creating a blank record, 4) Selecting a new field and subfield, and 5) Adding data to the field/subfield.

To create a new MARC file, first click on the Open button that is located in the MARC File area of J-MARC. This will display the Open file dialog box which is specific to the operating system that J-MARC is running on (Windows, in this example). Then, simply enter the name of the file in the File Name field and click on the Open button. Figure 3 shows the creation of a new MARC file called new.marc. After creating the new file, it must be opened as an existing file. This is accomplished by clicking on Open in the J-MARC window, and selecting the newly created file from the Open file dialog box. The file name will be displayed in the MARC file area.

After a new MARC file is open, records are added one at a time by click on the New button, which is located in the Record Modification and Navigation area of J-MARC. Also displayed in this area are the current record being created or edited, and the total number of records in the file. Figure 4 depicts the file new.marc with one record. The display of 1/1 indicates that the first record is being edited, and that the file contains one record.

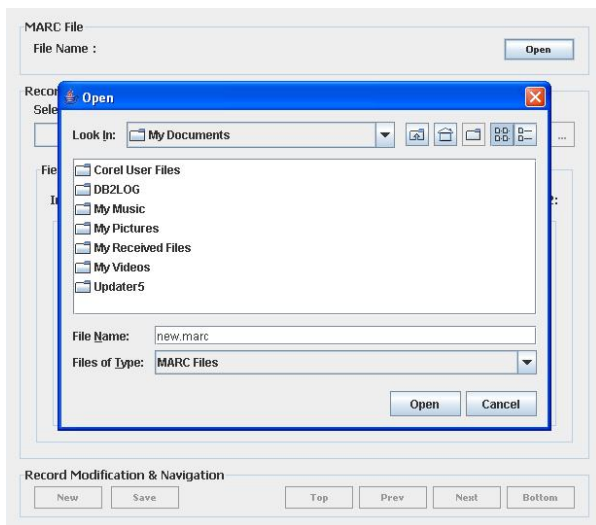


Figure 3. Specifying Name for New MARC File

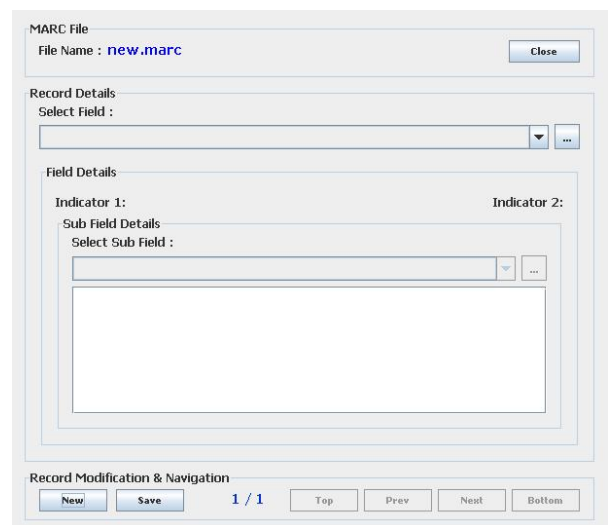


Figure 4. Open File with Blank Record

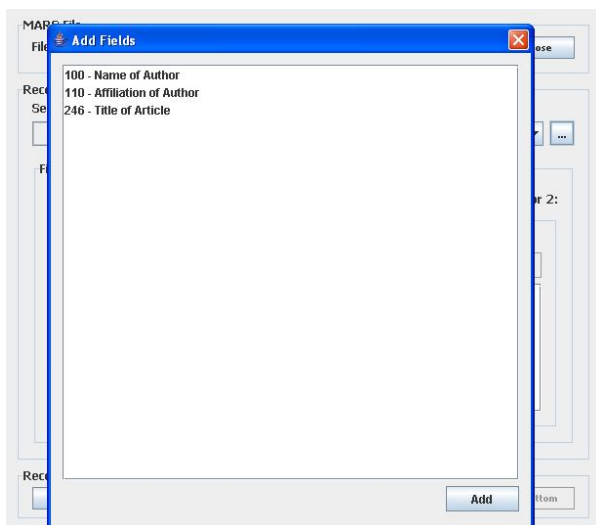


Figure 5. Selecting a Field

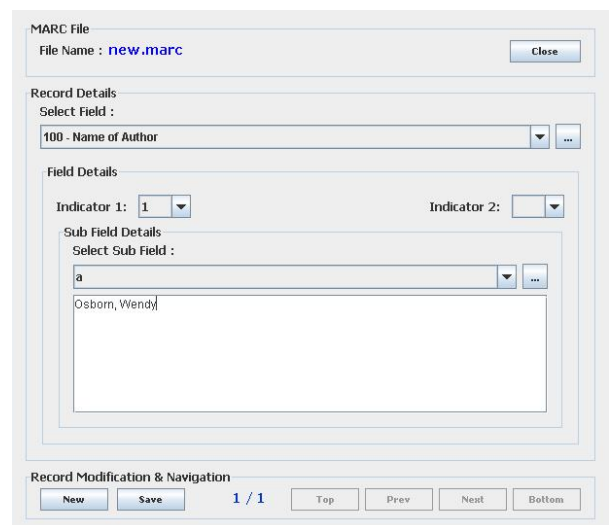


Figure 6. Adding Data to Field

After a new MARC record is created, fields are added one at a time. By clicking on the add field button (the button to the right of the Select Field drop-down box), a list of available fields are displayed to the user. There are two things to note here. First, only the permitted fields are displayed. Second, if a permitted field has a customized label, this is displayed to the user. From here, the user selects a field to add to the current MARC record and clicks on the Add button. Figure 5 depicts a list of three fields. Notice that each field has a customized label that is more suited to the community of users.

After a new field is selected, the user can select the appropriate indicators and subfields before entering data for the field. Figure 6 depicts the selected indicators, subfields and data for MARC field 100. After entering data for the

chosen field, the record can be saved by clicking on the Save button, which is located in the Record Modification and Navigation area of J-MARC. It should be noted that the user can add as many fields as he or she wishes before saving the record.

The user can add more records and customize them with the appropriate fields in the same manner. Then finished with the file, the user can close the MARC record by clicking on the Close button, located in the MARC file area. Later, if more records are to be added to the file, the file can simply be opened again. New records can be added, which are appended to the end of the MARC file. In addition, existing records can be modified, in terms of current content and by adding new fields and subfields.

5 Example: SAIR

Here, we provide an example of an existing digital library collection where a diverse group of users contribute MARC records, and how J-MARC can extend the functionality of the software used for maintaining the collection.

The Southern Alberta Information Resources (SAIR) project [5] is an online bibliography of resources related to the past and present of southern Alberta, Canada. It maintains MARC records on communities, regions, environmental issues, and international border issues, to name a few. The objectives of SAIR are:

1. To maintain an archive of all historical and current resources relevant to southern Alberta. These resources will also be digitized and made available for future generations to study and appreciate.
2. To foster collaboration between libraries, local community groups and other institutions. Some include historical societies, genealogical societies, and the local post-secondary institutions.

Different organizations involved in this collaborative effort run different operating systems. Therefore, a solution that supports these different platforms is required. In addition, many users involved in this effort are end-users that have limited computer skills. Therefore, a tool that is simple to use and understand is required.

SAIR is maintained using Greenstone [3]. As mentioned previously, Greenstone can process an existing file of MARC records. Through the GLI, one can edit records that are already imported by using a text editor, although one is editing a display version of MARC. Also, Greenstone can be customized to nicely display the records. However, it cannot create new MARC records, nor can it edit MARC records that are not imported into the collection. Therefore, additional functionality is required, in particular for creating new MARC records and files. To accomplish this, one can add the additional functionality to Greenstone itself. Another solution is to complement the existing software by adding a standalone MARC editor.

Therefore, by adding J-MARC to the system, users can create a MARC file that contains one or more records. Then, the MARC records can be imported into the SAIR library using the existing mechanisms provided by Greenstone. In addition, the newly created MARC files can be shared with other digital libraries that support the MARC format.

6. Review of Existing Tools

Many tools exist today that support the creation and editing of MARC records [6]. They can be classified into the following categories:

- Integrated tools. These tools provide full editing capabilities for MARC records. However, they are built

into an existing system. Therefore, they are not independent tools that can be used in conjunction with any digital library. Some record editing tools under this category include the following. BookWhere [7] and IsisMarc [8] are among the many tools that are available exclusively for the Windows operating system. Other systems, such as Millenium [9], TeraText [10], Unicorn [11], and JavaISIS [12] are written in Java. Therefore, these tools can be used across different systems.

- Standalone tools. These tools also provide full MARC record editing capabilities. In addition, they are completely independent and can be used in conjunction with any existing digital library. Some of these tools include MarcEdit [13], AGent MARCit [14] and Visual MARC Editor [15], which are among many standalone tools that are work exclusively on Windows.
- Application Programming Interfaces (APIs). APIs provide a set of functions for the creation of MARC tools that suit specific needs. One such tool is MARC4J [16], an API that is written in Java, and therefore is platform independent after it is used to implement a software tool.

J-MARC falls into the category of standalone tools. It differs from existing tools in the following ways:

- Digital Library Independent. As mentioned above, some cross-platform systems that provide MARC record editing capabilities are available. However, the record editor is integrated into a complete system, and cannot be removed and used with other existing systems. One solution could be to create new records in these systems and then export the records. However, this requires more work than is necessary to manipulate MARC records. In addition, it requires more software overhead than is necessary. J-MARC is an independent tool that can be used easily in conjunction with any existing digital library software package.
- Cross-platform support. Based on our findings, most standalone MARC record editing tools are written exclusively for the Windows family of operating systems. These tools cannot be run on other systems such as MAC OS X or Unix/Linux without the setup and use of emulation tools. It is possible to create a cross-platform standalone tool using the MARC4J API. However, this requires programming knowledge that many users may not have, and should not be required to learn. J-MARC can run on all major platforms and does not require programming knowledge to create or use. The only requirement for using J-MARC is a simple installation of the Java Runtime Environment (JRE).
- Simpler interface. Most existing MARC editing tools have cumbersome interfaces. J-MARC keeps editing

simple by only displaying one field at a time and allowing users to hide this field when not in consideration. In addition, the corresponding field name is also displayed upfront, which removes any confusion about what each field code represents.

7. Conclusion

We present J-MARC, which is a collaborative, cross-platform software tool for working with MARC bibliographic files. J-MARC can be used to create or edit a MARC file that contains one or more records. It is platform independent. It is a standalone program and therefore can be used with any library system that supports the MARC standard. Currently, we use J-MARC with the Greenstone digital library software. J-MARC is available as an open-source option under the terms of the GNU Public License. It can be obtained from:

<http://www.sadl.uleth.ca/jmarc/>

Future work includes the following. First, J-MARC performs little validation of the data entered by the user, and performs no validation for field-subfield relationships. This functionality will be added. Second, other functionality, such as deletion and a preview window will be added. Finally, because J-MARC is designed with modularity in mind, it can be extended easily to support the creation and manipulation of other bibliographic formats. One such format of interest is MarcXML [17]. We expect that the MarcXML format, among others, will be supported in the future.

References

- [1] The Library of Congress, *MARC Standards*, <http://www.loc.gov/marc> [Accessed 1 April 2007].
- [2] Witten, I.H. and Bainbridge, D. *How to Build a Digital Library*, Morgan Kauffman, 2003.
- [3] The New Zealand Digital Library Project, *The Greenstone Digital Library Software*, <http://www.greenstone.org> [Accessed 1 April 2007].
- [4] The Dublin Core Metadata Initiative, *The Dublin Core Metadata Element Set*, <http://www.dublincore.org/documents/dces/> [Accessed 1 April 2007].
- [5] Crewdson, K., Dew, I., and Osborn, W. "Southern Alberta Information Resources (SAIR) project update", *Newsletter of the Library Association of Alberta*, 2006, 151:16.
- [6] The Library of Congress, *MARC Tools*, <http://loc.gov/marc/marctools.html> [Accessed 1 April 2007].
- [7] Webclarity Software, *BookWhere*, <http://www.webclarity.info/products/bookwhere> [Accessed 1 April 2007].
- [8] UNESCO, *IsisMarc*, <http://portal.unesco.org/> [Accessed 1 April 2007].
- [9] Innovative Interfaces, *The Millenium Library System*, <http://www.iii.com/mill/index.shtml> [Accessed 1 April 2007].
- [10] TeraText Solutions, *The TeraText Document Management System*, <http://www.teratext.com/> [Accessed 1 April 2007].
- [11] SirsiDynix, *Unicorn Library Management System*, <http://www.sirsidynix.com/> [Accessed 1 April 2007].
- [12] Enea, R., *JavaISIS*, <http://portal.unesco.org/> [Accessed 1 April 2007].
- [13] Reese, T., *MarcEdit Record Editing Utility*, <http://oregonstate.edu/~reese/marcedit/> [Accessed 1 April 2007].
- [14] Auto-Graphics Inc., *AGent MARCIt Cataloguer*, <http://www.auto-graphics.com/> [Accessed 1 April 2007].
- [15] Algorithms Consultants Pvt., *Visual MARC Editor*, <http://www.slimpp.com/SlimPPsite/vismarc/> [Accessed 1 April 2007].
- [16] Peters, B., *MARC4J*, <http://marc4j.tigris.org> [Accessed 1 April 2007].
- [17] The Library of Congress, *The MARC 21 XML Schema*, <http://www.loc.gov/standards/marcxml/> [Accessed 1 April 2007].