

ABSTRACT

Editing bibliographic data is an important part of library information systems. In this paper we discuss existing approaches in developing user interfaces for editing MARC records. There are two basic approaches: screen forms that support entering bibliographic data without knowledge of the MARC structure, and direct editing of MARC records shown on the screen. This paper presents the Eclipse editor, which fully supports editing of MARC records. It is written in Java as an Eclipse plug-in, so it is platform-independent. It can be extended for use with any data store. The paper also presents a Rich Client Platform (RCP) application made of a MARC editor plug-in, which can be used outside of Eclipse. The practical application of the results is integration of the RCP application into the BISIS library information system.

INTRODUCTION

An important module of every library information system (LIS) is one for editing bibliographic records (i.e., cataloguing). Most library information systems store their bibliographic data in a form of MARC records. Some of them support cataloging by direct-editing of MARC record; others have a user interface that enables entering bibliographic data by a user who knows nothing about how MARC records are organized.

The subject of this paper is user interfaces for editing MARC records. It gives software requirements and analyzes existing approaches in this field. As the main part of the paper, we present the Eclipse editor for MARC records, developed at the University of Novi Sad, as a part of the BISIS library information system. Eclipse uses the MARC 21 variant of the MARC format.

The remainder of this paper describes the motivation for the research, presents the software requirements for cataloging according to MARC standards, and provides background on the MARC 21 format. It also describes the development of the BISIS software system, reviews the literature concerning tools for cataloging, and analyzes existing approaches in developing user interfaces for editing MARC records. The results of the research are presented in the final section, which describes the functionality and technical characteristics of the Eclipse MARC editor. The Rich Client Platform (RCP) version of the editor, which can be used independently of Eclipse, is also presented.

MOTIVATION

The motivation for this paper was to provide an improved user interface for cataloging by the MARC standard that will lead to more efficient and comfortable work for catalogers.

Bojana Dimić Surla (bdimic@uns.ns.ac.yu) is an Associate Professor, University of Novi Sad, Serbia.

There are two basic approaches in developing user interfaces for MARC cataloging. The first approach includes using a classic screen form made of text fields and labels with the description of the bibliographic data, without MARC standard indication. The second approach is direct editing of a record that is shown on the screen. Those two approaches will be discussed in detail in “Existing Approaches in Developing User Interfaces for Editing MARC Records” below. The current editor in the BISIS system is a mixture of these two approaches—it supports direct editing, but data input is done via text field, which opens on double click.¹

The idea presented in this paper is to create an editor that overcomes all drawbacks of previous solutions. The approach taken in creating the editor was direct record-editing with real-time validation and no additional dialogs.

Software Requirements for MARC Cataloging

The user interface for MARC cataloging needs to support following functions:

- Creating MARC records that satisfy constraints proposed by the bibliographic format
- Selecting codes for field tags, subfield names, and values of coded elements, such as character positions in leader and control fields, indicators, and subfield content
- Validating entered data
- Access to data about the MARC format (a “user manual” for MARC cataloging)
- Exporting and importing created records
- Providing various previews of the record, such as catalog cards

BACKGROUND

MARC 21

As was previously mentioned, the Eclipse editor uses the MARC 21 variant. MARC 21 consists of five formats: bibliographic data, authority data, holdings data, classification data, and community information.²

MARC 21 records consist of three parts: record leader, set of control fields, and set of data fields. The record leader content, which follows the *LDR* label, includes the logical length of the record (first five characters) and the code for record status (sixth character). After the record leader, there are control fields. Every control field is written in new line and consists of the three-character numeric tag and content of the control field. The content of the control field can be a single datum or a set of fixed-length bibliographic data. Control fields are followed by data fields in the record. Every line in the record that contains a data field consists of a three-character numeric tag, the value for the first and the second indicator—or the number sign (#) if indicators are not defined for the field—and the list of subfields that belong to the field.

Detailed analysis of MARC 21 shows that there are some constraints on the structure and content of the MARC 21 record. Constraints on the structure define which fields and subfields can appear more than once in the record (i.e., are the fields and subfields repeatable or not), the allowed length of the record elements, and all the elements of the record defined by MARC 21. Constraints on the record content are defined on the content of the leader, indicators, control fields and subfields. Moreover, some constraints connect more elements in the record (when the content of one element depends on the content of the other element in the record).

An example of constraint on the structure for data field 016 is that the field has the first indicator whereas the second indicator is undefined. The field 016 can have subfields a, z, 2, and 8, of which z and 8 are repeatable.

BISIS

The results presented in this paper belong to the research on the development of the BISIS library information system. This system, which has been in development since 1993, is currently in its fourth version. The editor for cataloging in the current version of BISIS was the starting point for the development of Eclipse, the subject of this paper.³

Apart from an editor for cataloging, the BISIS system has a module for circulation and an editor for creating Z39.50 queries.⁴ The indexing and searching of bibliographic records was implemented using the Lucene text server.⁵ As a part of the editor for cataloging, we developed the module generating various reports and catalog cards from MARC records.⁶ BISIS also supports creating an electronic catalog of UNIMARC records on the web, where the input of bibliographic data can be done without knowing UNIMARC but the entered data are mapped to UNIMARC and stored in the BISIS database.⁷

The recent research within the BISIS project relates to its extension for managing research results at the University of Novi Sad. For that purpose, we developed the Current Research Information System (CRIS) on the recommendation of the nonprofit organization euroCRIS.⁸ The paper “CERIF Compatible Data Model Based on MARC 21 Format” gives the proposal for the Common European Research Information Format (CERIF), a compatible data model based on MARC 21. In this model, a part of the CERIF data model that relates to research results is mapped to MARC 21. Furthermore, on the basis of this model, research management at the University of Novi Sad was developed.⁹ The paper “CERIF Data Model Extension for Evaluation and Quantitative Expression of Scientific Research Results” explains the extension of CERIF for evaluation of published scientific research. The extension is based on the semantic layer of CERIF, which enables classification of entities and their relationships by different classification schemas.¹⁰

The current version of the BISIS system is based on a variant of the UNIMARC format. The development of the next version of BISIS, which will be based on MARC 21, is in progress. The first task was migrating existing UNIMARC records.¹¹ The second task is developing the editor for MARC 21 records, which is the subject of this paper.

Cataloging Tools

An editor for cataloging is a standard part of a cataloger's workstation and the subject of numerous studies. Lange describes the cataloging development process from handwritten cataloging cards, to typewriters (first manual then electronic), to the appearance of MARC records and PC-based cataloger's workstations.¹²

Leroya and Thomas debate the influence of web development on cataloging. They stress that the availability of information on the web, as well as the possibility that more applications can be opened in the same time in different windows, greatly influence the process of creating bibliographic records. Their paper also indicates that there are some problems that result from using large numbers of resources from the web, such as errors that arise from copy-paste methods. Consequently, there is a need for automatic check of spelling errors and the possibility of a detailed review by a cataloger during editing.¹³

Khurshid deals with general principles of the cataloger's workstation, its configuration, and its influence on a cataloger's productivity. In addition to efficient access to remote and local electronic resources, Khurshid includes record transfer through a network and sophisticated record editing as important functions of a cataloger's workstation. Furthermore, Khurshid says it is possible to improve cataloging efficiency in the Windows-based cataloger's workstation by finding bibliographic records in other institutions and cutting and pasting lengthy parts of the record (such as summary notes) to their own catalog.¹⁴

Existing Approaches in Developing User Interfaces for Editing MARC Records

The basic source for this analysis of existing user interfaces for editing MARC records was the official site for MARC standards of the Library of Congress in addition to scientific journals and conferences. The analysis of existing systems shows that there are two basic approaches in the implementation of editing MARC records:¹⁵

- Entering bibliographic data in classic screen forms made of text fields and labels, which does not require knowledge of the MARC format (Concourse,¹⁶ Koha,¹⁷ J-MARC¹⁸)
- Direct editing of a MARC record shown on the screen (MARCEdit,¹⁹ IsisMARC,²⁰ Catalis,²¹ Polaris,²² MARCMaker and MARCBraker,²³ ExLibris Voyager²⁴).

Both of these approaches have advantages and disadvantages. The drawback of the first approach is that it provides a limited set of bibliographic data to edit, and the extension of that set implies changes to the application, or in the best cases changes in configuration. Another problem is that there are usually a lot of text fields, text areas, combo boxes, and labels on the screen that need to be organized into several tabs or additional windows. This situation usually makes it difficult for the users to see errors or to connect different parts of the record when checking their work. Moreover, all found solutions from the first group perform little validation of data entered by the user.²⁵ One important advantage of the first approach is that the application can be used by a user

who is not familiar with the standard, thus the need for access to MARC data can be avoided (one of functions listed “MARC 21” above).

As for second approach, editing a MARC record directly on the screen overcomes the problem of extending the set of bibliographic data to enter. It also enables users to scan entered data and check the whole record, which appears on the screen. Users can also copy and paste parts of records from other resources into the editor.

However, a majority of those applications are actually editors for editing MARC files that are later uploaded in some database or transformed in some other format (Marcredit, MARCmaker and MARCbreaker, Polaris), and they usually support little or no data validation.²⁶ They allow users to write anything (i.e., the record structure is not controlled by the program), and only validate at the end of the process when uploading or transforming the record. Among those editors there are those, such as Catalis and IsisMARC, that present the MARC record as a table. They support the control of structure, but the record presented in this way is usually too big to fit on the screen, so it is separated into several tabs.

An important function of editing MARC records is selecting code for coded elements that can be positioned in the leader or control field, value of the indicator, or value of the subfield. There are also field tags or subfield codes that sometimes need to be selected for addition to a record. All analyzed editors provide additional dialogs for picking this code that require the user to constantly open and close dialogs, which sometimes can be annoying for the user.

One important fact about editors in the second group is that they can be used only by a user who is familiar with MARC, so access to the large set of MARC element descriptions can make the job easier. Some of the mentioned systems provide descriptions of the fields and subfields (e.g., IsisMARC), but most of them do not.

FINDINGS

The editor for MARC records was developed as a plug-in for Eclipse; therefore it is similar to Eclipse’s Java code editors. As the editor is written in Java, it is platform-independent. The main part of this editor was created using oAW Xtext framework for developing textual domain-specific languages.²⁷ It was created using model-driven software development by specifying the model of MARC record in a form of Xtext grammar and generating the editor. All main characteristics of the editor were generated on the basis of the specification of constraints and extensions of the Xtext grammar—therefore all changes to the editor can be realized by changing the specification.

Moreover, this editor can be easily adjusted for any database by using the concept of extension and extension point in the Eclipse plug-in. We make this application independent of Eclipse by using Rich Client Platform (RCP) technology. This editor is implemented for MARC 21 bibliographic and holdings formats.

User Interface

Figure 1 shows the editor opened within Eclipse. The main area is marked with “1”—it shows the MARC 21 file that is being edited. That file contains one MARC 21 bibliographic record. The tags of the fields and subfields codes are highlighted in the editor, which contributes to presentation clarity. The area marked with “2” serves for listing the errors in the record, that is, nonvalid elements entered in the record. The area marked with “3” shows data about MARC 21 in a tree form. This part of the screen has two other possible views: a MARC 21 holdings format tree and a navigator, which is the standard Eclipse view for browsing resources for the opened project. The actions available for creating a record are available in the cataloging menu and on the cataloging toolbar, which is marked with “4.” These are actions for previewing the catalog card, creating a new bibliographic record, loading a record from a database (importing the record), uploading a record to a database (exporting the record), and creating a holdings record for this bibliographic record.

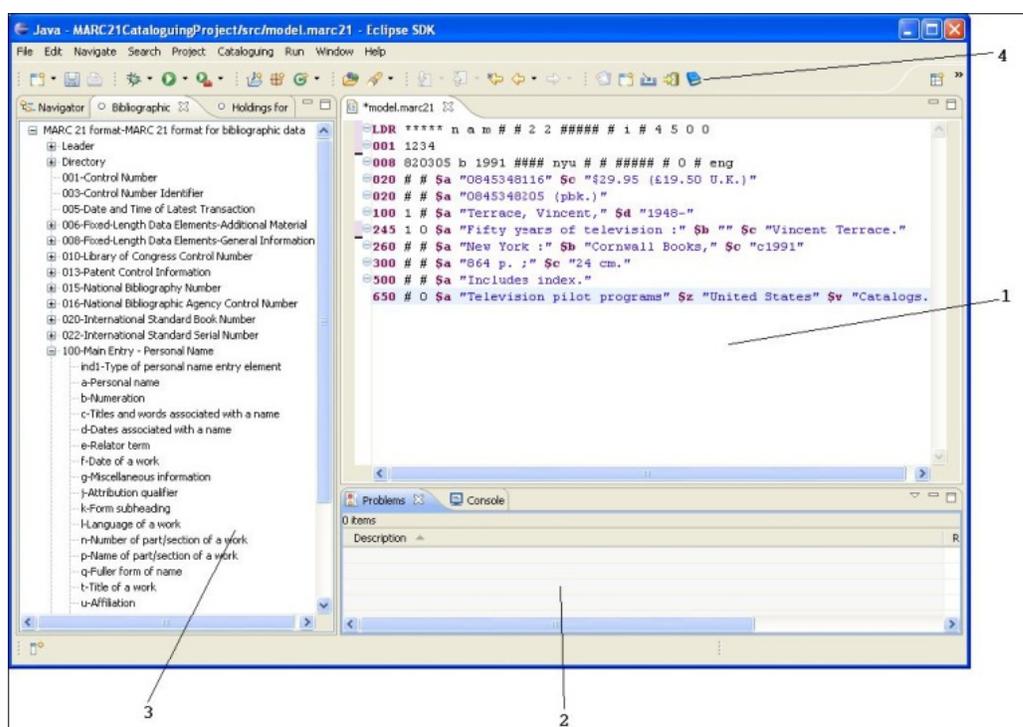


Figure 1. Eclipse Editor for MARC Records

In the Eclipse editor for MARC, selecting codes is enabled without opening additional dialogs or windows (figure 2). That is a standard Eclipse mechanism for code completion: typing Ctrl + Space opens the dropdown list with all possible values for the cursor's current position.

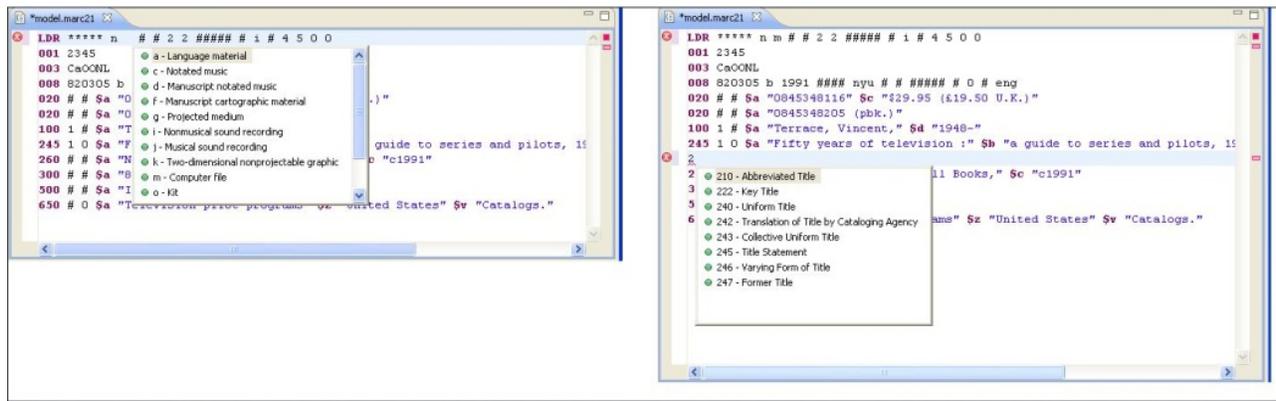


Figure 2. Selecting Codes

Record validation is done in real time, and every violation is shown while editing (figure 3). Figure 3 depicts two errors in the record: one is a wrong value in the second character position in control field 008, and another is that two 100 fields were entered, which is a field that cannot be duplicated in a record.

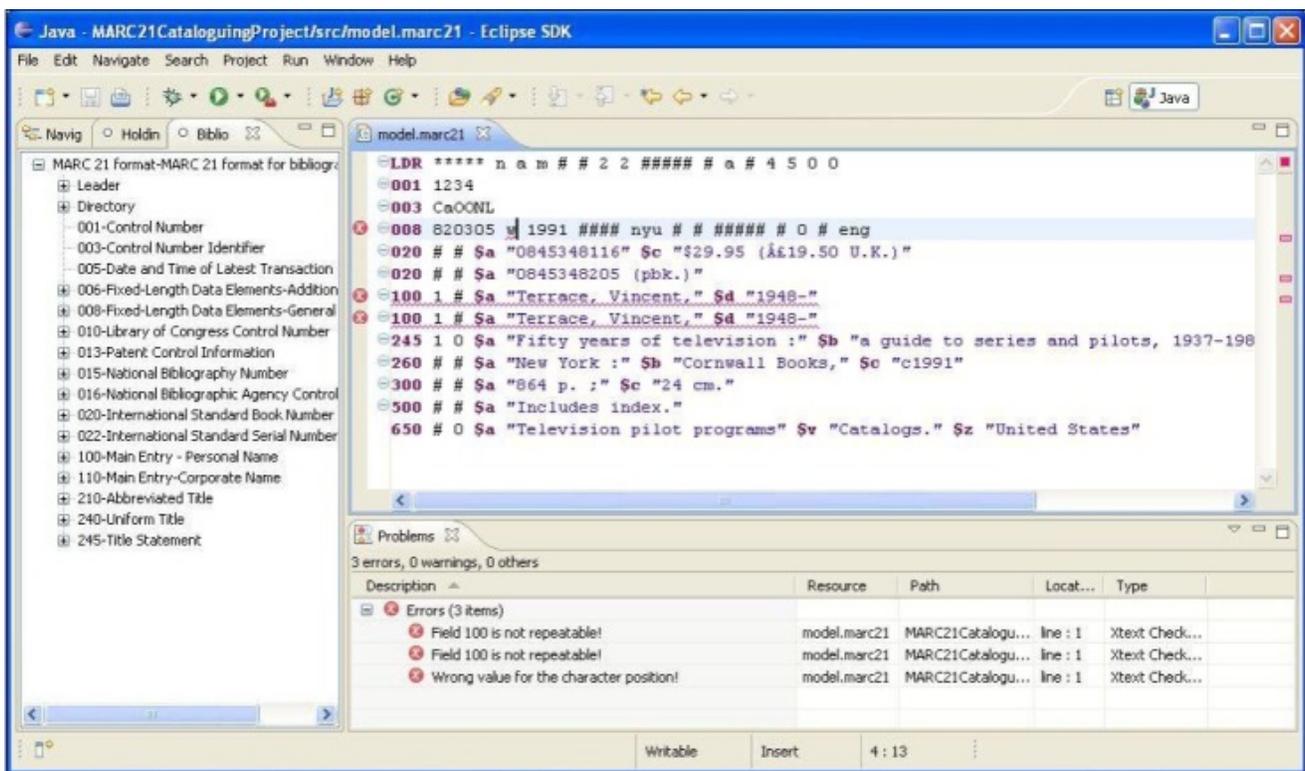


Figure 3. Validation Errors

RCP Application of the Cataloging Editor

As shown above, the editor is available as an Eclipse plug-in, which raises the question of what a cataloger will do with all the other functions of the Eclipse Integrated Development Environment (IDE). As seen in figures 1 and 3, there are a lot of additional toolbars and menus that not related

to cataloging. The answer lies in RCP technology. RCP technology generates independent software applications on the basis of a set of Eclipse plug-ins.²⁸

The main window of an RCP application with additional actions is shown in figure 4. Beside the Cataloguing menu that is shown, the window also contains the File menu, which includes Save and Save As actions, as well as the Edit menu, which includes Undo and Redo actions. All of these actions are also available via the toolbar.

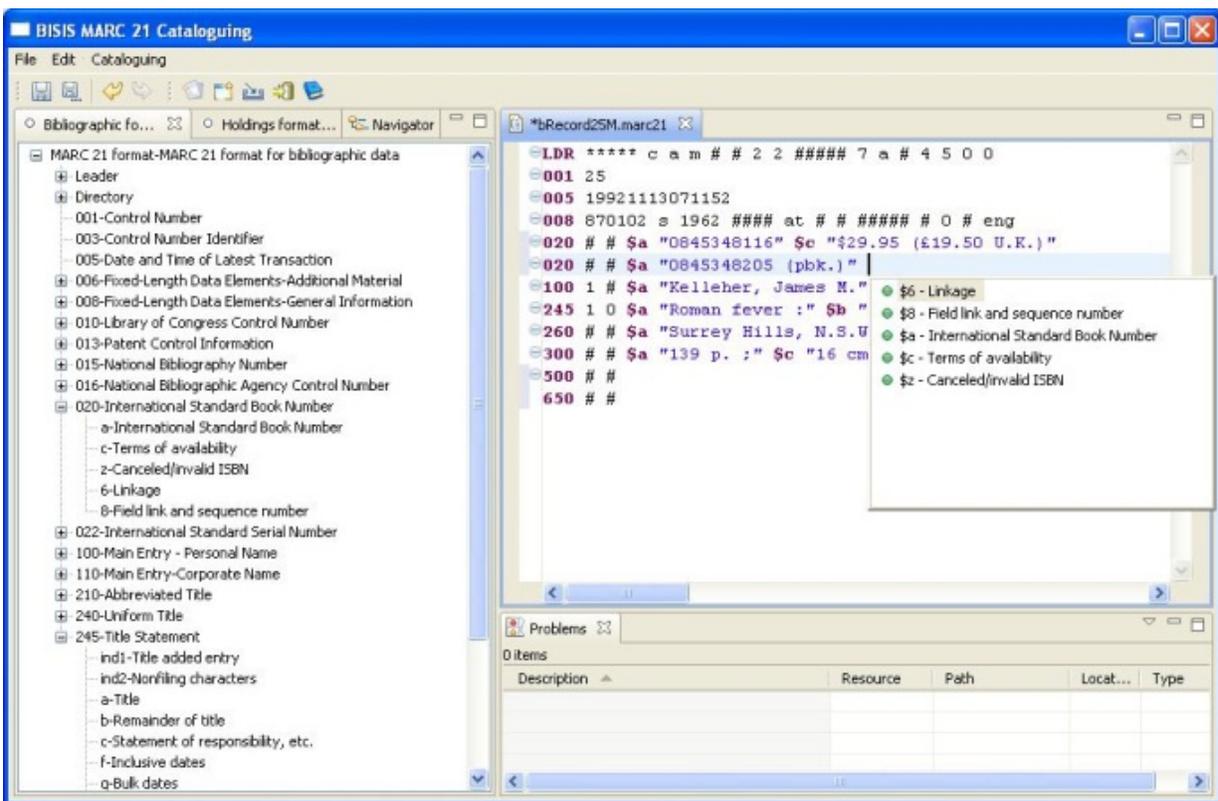


Figure 4. RCP Application

CONCLUSION

The goal of this paper was to review current user interfaces for editing MARC records. We presented two basic approaches in this field and analyzed of advantages and disadvantages of each. We then presented the Eclipse MARC editor, which is part of the BISIS library software system.

The idea behind Eclipse is inputting structured MARC data in the form similar to programming language editors. The author did not find this approach in the accessible literature.

The RCP application of the presented editor will find its practical application in future versions of the BISIS system. It represents an upgrade of the existing editor and a starting point for forming the version of the BISIS system that will be based on MARC 21. The acquired results can also be

used for the input of other data into the BISIS system, including data from the CRIS system used at the University of Novi Sad.

This paper shows that Eclipse plug-in technology can be used for creating end user applications. The development of applications with the plug-in technology enables the use of a big library of created components from the Eclipse user interface, whereby writing source code is avoided. Additionally, the plug-in technology enables the development of extendible applications by using the concept of the extension point. In this way, we can create software components that can be used by a great number of different information systems. By using the concept of “extension point,” the editor can be extended by the functions that are specific for a data store. An extension point was created for export and import of MARC records, which means the MARC editor plug-in can be used with any database management system by extending this extension point in Eclipse plug-in technology.

Future work in the development of the Eclipse MARC editor is to implement support for additional MARC formats, for authority and classification data, and for community information. These formats propose the same record structure but have different constraints on the content and different sets of fields and subfields, as well as different codes for character positions and subfields. Therefore the appearance of the editor will remain the same. The only difference will be the specification of the constraints and codes for code completion.

Another interesting topic for discussion is considering implementation of other modules of library information systems in Eclipse plug-in technology.

REFERENCES

1. Bojana Dimić and Dušan Surla, “XML Editor for UNIMARC and MARC21 cataloging,” *Electronic Library* 27 (2009): 509–28; Bojana Dimić, Branko Milosavljević, and Dušan Surla, “XML Schema for UNIMARC and MARC 21 Formats,” *Electronic Library* 28 (2010): 245–62.
2. Library of Congress, “MARC Standards,” <http://www.loc.gov/marc> (access February 19, 2011).
3. Dimić and Surla, “XML Editor,” Dimić, Milosavljević, and Surla, “XML Schema.”
4. Danijela Tešendić, Branko Milosavljević, and Dušan Surla, “A Library Circulation System for City and Special Libraries,” *Electronic Library* 27 (2009): 162–68; Branko Milosavljevic and Danijela Tešendić, “Software Architecture of Distributed Client/Server Library Circulation,” *Electronic Library*, 28 (2010): 286–99; Danijela Boberić and Dušan Surla, “XML Editor for Search and Retrieval of Bibliographic Records in the Z39.50 Standard,” *Electronic Library* 27 (2009): 474–95.
5. Branko Milosavljević, Danijela Boberić, and Dušan Surla, “Retrieval of Bibliographic Records Using Apache Lucene,” *Electronic Library* 28 (2010): 525–36.

-
6. Jelana Rađenović, Branko Milosavljević, and Dušan Surla, "Modelling and Implementation of Catalogue Cards Using FreeMarker," *program: Electronic Library and Information Systems* 43 (2009): 63–76.
 7. Katarina Belić and Dušan Surla, "Model of User Friendly System for Library Cataloging," *ComSIS* 5 (2008): 61–85; Katarina Belić and Dušan Surla, "User-Friendly Web Application for Bibliographic Material Processing," *Electronic Library* 26 (2008): 400–410; EuroCRIS homepage, www.eurocris.org (accessed February 21, 2011).
 8. Dragan Ivanović, Dušan Surla, and Zora Konjović, "CERIF Compatible Data Model Based on MARC 21 Format," *Electronic Library* 29 (2011).
<http://www.emeraldinsight.com/journals.htm?articleid=1906945>.
 9. euroCRIS, "Common European Research Information Format,"
<http://www.eurocris.org/Index.php?page=CERIFreleasesandt=1> (accessed February 21, 2011); Dragan Ivanović et al., "A CERIF-Compatible Research Management System Based on the MARC 21 Format," *program: Electronic Library and Information Systems* 44 (2010): 229–51.
 10. Gordana Milosavljević et al., "Automated Construction of the User Interface for a CERIF-Compliant Research Management System," *The Electronic Library* 29 (2011).
<http://www.emeraldinsight.com/journals.htm?articleid=1954429>; Dragan Ivanović, Dušan Surla, and Miloš Racković, "A CERIF Data Model Extension for Evaluation and Quantitative Expression of Scientific Research Results," *Scientometrics* 86 (2010): 155–72.
 11. Gordana Rudić and Dušan Surla, "Conversion of Bibliographic Records to MARC 21 Format," *Electronic Library* 27 (2009): 950–67.
 12. Holley R. Lange, "Catalogers and Workstations: A Retrospective and Future View," *Cataloging & Classification Quarterly* 16 (1993): 39–52.
 13. Sarah Yoder Leroya and Suzanne Leffard Thomas, "Impact of Web Access on Cataloging," *Cataloging & Classification Quarterly* 38 (2004): 7–16.
 14. Zahirrudin Khurshid, "The Cataloger's Workstation in the Electronic Library Environment," *Electronic Library* 19 (2001): 78–83.
 15. Library of Congress, "MARC Standards," <http://www.loc.gov/marc> (access February 19, 2011).
 16. Book Systems, "Concourse Software Product,"
<http://www.booksys.com/v2/products/concourse> (accessed February 19, 2011).
 17. Koha Library Software Community homepage, <http://koha-community.org> (accessed February 19, 2011).

-
18. Wendy Osborn et al., "A Cross-Platform Solution for Bibliographic Record Manipulation in Digital Libraries," (paper presented at the Sixth IASTED International Conference Communications, Internet and Information Technology, July 2–4, 2007, Banf, Alberta, Canada).
 19. Terry Reese, "MarcEdit—Your Complete Free MARC Editing Utility," <http://people.oregonstate.edu/~reese/marcedit/html/index.php> (accessed February 19, 2011).
 20. United Nations Educational Scientific and Cultural Organization, "IsisMARC," http://portal.unesco.org/ci/en/ev.php-URL_ID=11041&URL_DO=DO_TOPIC&URL_SECTION=201.html (accessed February 19, 2011).
 21. Fernando J. Gómez "Catalis," <http://inmabb.criba.edu.ar/catalis> (accessed February 19, 2011).
 22. Polaris Library Systems homepage, <http://www.gisinfosystems.com> (accessed February 19, 2011).
 23. Library of Congress, "MARCMaker and MARCBreaker User's Manual," <http://www.loc.gov/marc/makrbrkr.html> (accessed February 19, 2011).
 24. ExLibris, "ExLibris Voyager," <http://www.exlibrisgroup.com/category/Voyager> (accessed February 19, 2011).
 25. Book Systems, "Concourse Software Product."
 26. Bonnie Parks, "An Interview with Terry Reese," *Serials Review* 31 (2005): 303–8.
 27. Eclipse.org, "XText," <http://www.eclipse.org/Xtext> (accessed February 19, 2011).
 28. The Eclipse Foundation, "Rich Client Platform," http://wiki.eclipse.org/index.php/Rich_Client_Platform (accessed February 19, 2011).