Notes on Operations Metadata Makeover

Transforming MARC Records Using XSLT

Violeta Ilik, Jessica Storlien, and Joseph Olivarez

Catalogers have become fluent in information technology such as web design skills, HyperText Markup Language (HTML), Cascading Stylesheets (CSS), eXensible Markup Language (XML), and programming languages. The knowledge gained from learning information technology can be used to experiment with methods of transforming one metadata schema into another using various software solutions. This paper will discuss the use of eXtensible Stylesheet Language Transformations (XSLT) for repurposing, editing, and reformatting metadata. Catalogers have the requisite skills for working with any metadata schema, and if they are excluded from metadata work, libraries are wasting a valuable human resource.

Being a cataloger requires more than knowledge and understanding Machine-Readable Cataloging (MARC) format, the Anglo American Cataloging Rules (AACR2), the Library of Congress Descriptive Cataloging Manual (DCM), Library of Congress' Subject Headings Manual (SHM), Library of Congress Classification (LCC), Resource Description and Access (RDA), and other cataloging rules and standards. Cataloging practices must also embrace the opportunity to employ new schemas for resource description and how to reuse and repurpose existing metadata.

In the current library ecosystem, catalogers must be willing to assume new responsibilities to enable information to be organized, repurposed, and shared with patrons and other libraries. A large part of these new responsibilities are grounded in the importance and use of metadata to meet the needs of libraries, including creating interoperable data, repurposing data, and building digital repositories. Catalogers have the fundamental skills to successfully work with and repurpose metadata. These skills include, but are not limited to, organization of information, knowledge of commonly used access points, and a growing knowledge of information technology. Catalogers must also develop fluency in information technology (IT) including HyperText Markup Language (HTML), Cascading Stylesheets (CSS), eXensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), and MARCXML (an XML schema based on MARC21) to expand and reimagine their work. By encouraging catalogers to work with metadata creation and standards and to learn web development skills, libraries are using their resources and staff efficiently. This paper will explain how catalogers with intermediate knowledge of HTML, CSS, and XML can develop

Violeta Ilik (vilik@library.tamu.edu) is Assistant Professor, Semantic Technologies Librarian, Texas A&M University, College Station, Texas. **Jessica Storlien** (jessicahennen@gmail.com) is Reference and Instruction Librarian, Blinn College, Brenham, Texas. **Joseph Olivarez** (jolivare@library.tamu.edu) is a Library Specialist III, Texas A&M University.-

Manuscript submitted April 2, 2013; returned to authors for revision July 5, 2013; revisions submitted August 27, 2013; tentatively accepted pending minor revisions September 30, 2013; accepted for publication April 22, 2014.

The authors would like to express the deepest appreciation and gratitude to Nancy Burford, Veterinary Collectors Curator, for her continuous support and generous help through the process. The authors would also like to express special appreciation and thanks to Anne Highsmith, Director of Consortia Systems, for encouraging our research and for helping us write the XSLT for the first experiment.

stylesheets to transform or enhance XML documents.

Literature Review

A survey conducted in 2007 by Ma investigated how metadata was implemented in Association of Research Libraries (ARL) member libraries and revealed that the metadata qualifications and responsibilities required by most responding institutions included knowledge of MARC cataloging, advanced knowledge of metadata crosswalks, and that knowledge of XML and the Open Archives Initiative (OAI) were considered desirable.[1] Park, Lu, and Marion analyzed cataloging position descriptions for vacancies posted on the Autocat discussion list between 2005 and 2006.[2] Their results revealed that of the required qualifications, computer skills (including, but not limited to, hardware, software, Microsoft Office applications, word processing, spreadsheets, and Microsoft Windows) appeared in 32.1 percent of the postings. Metadata knowledge, including but not limited to, Dublin Core (DC), Encoded Archival Description (EAD), Metadata Object Description Schema (MODS), Text Encoding Initiative (TEI), and Visual Resources Association (VRA) appeared in 23.5 percent of the postings. Web knowledge, including but not limited to, the World Wide Web, HTML, Standard Generalized Markup Language (SGML), XML appeared in 16.3 percent of the postings. Results reveal that advances in technology have created a new realm of desired skills, qualifications and responsibilities for catalogers.

Hseih-Yee asserts that although catalogers may not be involved in writing programming codes, they need sufficient knowledge of the technologies and tools affecting information organization and services to communicate with vendors and systems management units.[3] The current trend is for catalogers to be involved in learning to code through various collaborative venues. Calhoun advocates for librarians to become IT-fluent to better support the future of library information dissemination.[4]

Reese maintains that as more institutions bring collections online, technical services staff will continue to face the growing issue of distributed metadata retrieval.[5] He further states that technical services departments have viewed metadata harvesting and transformation as the responsibility of library technology departments. He discusses Texas A&M University Libraries' method for metadata repurposing developed by Surratt and Hill in 2004 and notes that data conversion was moved outside the cataloging department, creating a barrier between catalogers and the developers of the script.[6] Reese is the creator of the free Windows-based MARC editing tool, MarcEdit (marcedit.reeset.net), which can be used in an everyday cataloger's work and provides a means for editing metadata, XML crosswalking, and metadata harvesting via the Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH). With this tool, Reese proves that new tools and workflows will continue to be developed and more technical services departments will turn to metadata harvest and capture as a viable method of generating metadata for digital collections.[7]

Tosaka stresses the importance of metadata transformation to enable reuse, and he cites the need for additional studies on metadata interoperability and crosswalks.[8] He also states that the ability to repurpose metadata is important due to the "increasingly global, interdisciplinary environment where users must deal with metadata records from multiple databases with their individual data structures."[9] Tosaka deduces that collections of malleable, shareable metadata are in demand, and this must be considered by current data-creation standards.[10] The workflow used by the University of Illinois Library "implements the eXtensible Markup Language and Open Archives Initiative-Protocol for Metadata Harvesting to create cataloging records for digitized books."[11] The author further makes the point that libraries have to develop these new workflows to be able to keep abreast of the abundance of digitized books produced by mass digitization projects.

Woodley provides these simple definitions for mapping and crosswalks: mapping compares and analyzes two or more metadata schemas while crosswalks are the product of the mapping process.[12] St. Pierre and LaPlant define crosswalks as the "specification for mapping one metadata standard to another."[13] Metadata crosswalks are central to ensuring seamless access to information from various systems and essential in converting data from one format to another. Woodley points out the enormity of the tasks involved in repurposing metadata, including transforming records from one schema to another and merging records created using different schemas or standards.[14] A similar study, conducted by Rudic and Surla, discusses "the application of the XML technologies for the conversion of the bibliographic records between the different bibliographic formats (YUMARC, UNIMARC and MARC 21)" used in the library system in Serbia.[15]

Before initiating a metadata crosswalk, it is important to have an awareness of common issues associated with metadata and crosswalks. According to Dushay and Hillman, the four categories of metadata problems are missing data, incorrect or erroneous data, confusing or inconsistent data, and insufficient data.[16] Woodley states that common issues with migrating data include ambivalent matches, hybrid bibliographic records, data mapping to multiple fields or combining into single fields during migration, orphaned data parsed into incongruous fields, mixed standards in

original data, MARC data loss during the migration, and flat structure versus hierarchical structures.[17] St. Pierre and LaPlant describe similar issues with metadata crosswalks, including reconciling metadata organization systems, choice of unanalogous processes during metadata standards creation, imprecise definitions or alternate naming choices that inhibit element-to-element mapping, information being lost or combined during mapping, and unharmonious hierarchical structures.[18] Godby, Young, and Childress note that a problem with crosswalks is they are not always identified as a standard, and point to the digital library community's opposing views on crosswalks.[19] One view maintains that crosswalks are a stopgap measure, a local and temporary solution, until a single data standard is developed. The other view asserts that crosswalks "represent an attempt to identify interoperable elements among standards"; this implies that crosswalks should become a standard practice.[20]

According to Chandler and Westbrooks, construction of open systems and methods that provide the ability to link between different types of metadata is the path to information discovery.[21] They also discuss the need for the library technical services departments to be proactive not only in the creation and maintenance of non-MARC metadata, but more importantly, in the development of a means for widely sharing metadata with libraries that require it for resource discovery and access. Ahronheim and Marko cite the need for catalogers to participate in both the development and use of descriptive standards because standards easily allow data to be reused.[22] Woodley states, "Consistently recorded, reliable metadata can be reused and combined with metadata records that have been created according to different standards to create richer, more informative information objects."[23] Calhoun concludes that metadata can and should be reused, and libraries must ensure interoperability of their metadata for this very reason.[24]

The appeal to create interoperable data has led to a discussion of using XML to manipulate data, including MARC data. Johnson ascertained that XML and HTML standards closely identify with web development, allowing for effective integration with web interfaces.[25] XML is inherently hierarchical, web-oriented, works well with web applications, and allows for retrieval and manipulation of data. XML is considered an unencumbered format that facilitates the ability to experiment with library data, for example, transforming MARCXML records into various other schemas. Johnson asserts "XML presents new opportunities and extended life for MARC" because XML can transfer MARC encoded information.[26] Due to the dual roles of librarians as information consumers and metadata producers, it seems natural that librarians would influence and participate in the development of XML software and applications. The authors of this paper propose that catalogers who are empowered by library administrators and are encouraged to assume new responsibilities can acquire skills such as programming languages and combine them with existing cataloging expertise to successfully implement metadata projects.

Using XML and XSLT

Catalogers at the Texas A&M University Libraries chose XSLT to manipulate XML records because of its relative ease of use. According to Keith, stylesheets are the ideal format for the maintenance of XML data transformations due to the native adaptability and simplicity of stylesheets.[27] No special software is needed to change stylesheets. He further states that because XSLT documents are simple text files, just like XML documents, they can be edited with word-processing software at the most basic level.

Understanding of HTML and CSS is essential to comprehending XML and XSLT. Tennison states that XML is a meta-markup language specifically designed for ease of use with the web that is human-readable and straightforward for applications to read and understand.[28] An XML document written in XSLT is commonly known as an XSLT stylesheet and is usually assigned the file extension .xsl.[29] Each XSLT stylesheet describes how a set of XML documents (the source documents) should be converted to other documents (the result documents), whether they are eXtensible Stylesheet Language Formatting Objects (XSL-FO), eXtensible HyperText Markup Language (XHTML), comma-delimited text, or any other text-based format, such as HTML. An XSLT stylesheet will typically take source documents written in one markup language and produce a result in another markup language, which can be used by a specialist application, such as XHTML, for presentation in a browser.[30]

To perform a transformation, XSLT must be capable of pointing to information in the source document to process the information and include it in the result. XML Path Language (XPath) serves as the guide for XSLT.[31] The most important role of XPath is to collect information from an XML document by navigating through the document. A good XML editor, such as oXygen, is necessary to experiment and work with XML and XSLT. The Library of Congress (LC) developed the MARCXML architecture and MARCXML toolkit to standardize the exchange of MARC structured data in XML. The core of the MARCXML framework is a simple XML schema that contains MARC data. As stated on the LC website, this base schema output can be used where full MARC records are needed or can act as a "bus" to enable MARC data records to go through further transformations,

such as to DC or processes like validation.[32] Control fields, including the leader, are treated as data strings, while variable fields are treated as elements and the tag and indicators are treated as attributes. Subfields are treated as sub-elements with the subfield code as an attribute.

The MARCXML schema provides easy access to discrete pieces of data, such as data stored in the leader, 008, and subfields, and enables the creation of XML stylesheets to manipulate and transform the data.[33] Information is accessed at the subfield level with simple XPath expressions. Many types of transformation exist, such as MARCXML to DC, MODS, or just simple style sheets that allow enhancement of MARCXML files. Keith states that the MARC21 file format is not easy to modify or transform to other schemas, and it is not common for a software developer to understand MARC.[34] A well-written specification is required to manipulate MARC metadata. Unlike MARC, XML is simple, although Coyle notes that it allows for the creation of complicated data records.[35] She explains XML by comparing it to MARC tags, such as the use of "245," which in XML would be written as <title>Title of the book</title>.

A major difference between MARC and XML is that XML uses brackets to denote the beginning tag <> and ending tags </>. The tags need to be predefined in a data-format-definition structure. XML is hierarchical, as are MARC tags and subfields. However, XML is potentially more hierarchical as there is no limit to the number of levels, unlike MARC, which is limited in number by the established standard.

Catalogers at Texas A&M University Libraries used the stylesheets available on LC's Network Development and MARC Standards Office site for the second experiment. As stated on the LC site, "this framework is intended to be flexible and extensible to allow users to work with MARC data in ways specific to their needs. The framework itself includes many components such as schemas, stylesheets, and software tools."[36] For the purpose of data transformation for sample records from MARCXML to DC, the authors adapted the LC MARCXML to Resource Description Framework (RDF) Encoded Simple DC Stylesheet. The use of this stylesheet produced the DC-RDF file format.

First Experiment: Creation of Bibliographic Records for Electronic Information Resources from Bibliographic Records for the Print Version of the Same Resource

The XSLT for this experiment was created by one of the three authors. As libraries frequently purchase electronic versions of resources owned in print, the authors believed it would save time and money to create MARC records for the electronic resources by reusing the existing metadata stored in the bibliographic records for the print versions. Using a simple XML editor, they created a stylesheet to transform bibliographic records for the print resources to bibliographic records for electronic version of same resources in MARCXML file format (see appendix A). The bibliographic records in MARCXML were transformed through the MarcEdit utility into MARC records ready for import into the local Integrated Library System (ILS).

The authors began by identifying fields in the original bibliographic records that would not be reused. There are several fields used in records for print resources that are unnecessary when describing electronic resources. Other fields that need to be removed are unique record identifiers, such as the OCLC number and the ILS bibliographic number. The command line for the removal of those fields is as follows:

```
<xsl:template match=
"marc:controlfield[@
tag='nnn']"/>
```

After deleting the unnecessary print record fields, the mandatory fields needed in MARC records for electronic resources, such as the 006 and 007 fields, were added.

```
<xsl:template name= "t006">
<marc:controlfield tag=
"006">
 <marc:subfield code= "a">m
o d</marc:subfield>
</marc:controlfield>
</xsl:template>
<xsl:template name="t007">
<marc:controlfield tag=
"007">
 <marc:subfield code=
"a">c</marc:subfield>
 <marc:subfield
code="b">r</marc:subfield>
</marc:controlfield>
</xsl:template>
```

A command for changing the 008 form of item position to "o" for online was also created.

```
<xsl:template name="t008">
<xsl:variable name= "f008"
select= "concat(substring(m
arc:controlfield[@tag='008'],
1, 23), 'o',substring(marc:co
ntrolfield[@tag='008'], 25))
"/>
<marc:controlfield tag=
"008">
 <xsl:value-of select=
"$f008/>
</marc:controlfield>
</xsl:template>
```

Other fields that are recommended for describing electronic books (e-books) were also added, such as the 300 field with the value of "1 online resource" in subfield a, the 588 field with a value of "Description based on print record," and the 655 field with a value in subfield a of "Electronic

books" and subfield "2" with a value of "local":

```
<xsl:template name= "t300">
<marc:datafield tag= "300"
ind1= "" ind2="">
 <marc:subfield code="a">
 <xsl:text>1 online
resource</xsl:text>
 </marc:subfield>
</marc:datafield>
</xsl:template>
<xsl:template name= "t588">
<marc:datafield tag= "588"
ind1= ""ind2="">
 <marc:subfield code="a">
 <xsl:text>Description based
on print record.</xsl:text>
 </marc:subfield>
</marc:datafield>
</xsl:template>
<xsl:template name= "t655">
<marc:datafield tag= "655"
ind1= ""ind2="7">
 <marc:subfield code= "a>
 <xsl:text>Electronic
books.</xsl:text>
 </marc:subfield>
 <marc:subfield code="2">
 <xsl:text>local</xsl:text>
 </marc:subfield>
</marc:datafield>
</xsl:template>
```

After transforming the MARCXML file and creating a new MARCXML file for e-books, the authors processed the file using MarcEdit software to convert it to a MARC file to import it to the local ILS or to the Online Computer Library Center (OCLC) (see figure 1). The process described above could allow a cataloging department to streamline the creation of thousands of bibliographic record for electronic resources from bibliographic records for print materials the library already holds in the ILS and OCLC.

The transformation made to the bibliographic records for print versions of information resources to bibliographic records for electronic information resources is shown in appendix B. There are two examples of bibliographic records in appendix B: one for a print information resource (record 1) and one for an electronic information resource (record 2). The changes made to the bibliographic records for print resources are made using the stylesheet referred to in appendix A.

The time used by one of the authors in creating the stylesheet for this first experiment was approximately ten hours over a period of three weeks. The initial learning curve needed for the first stylesheet accounts for time consumed in its creation. With each subsequent stylesheet creation, the time required lessened to a little more than thirty minutes. Once the stylesheet was formatted, we were able to create thousands of new electronic resource bibliographic records in seconds via the transformative ability of the process.

Second Experiment: Transforming Sample MARCXML to Dublin Core Records

The next experiment involved the transformation of XML documents from one schema to another using XSLT, specifically the transformation of MARCXML to DC XML. The authors adapted the existing MARCXML to an RDF Encoded Simple DC Stylesheet, available on the LC website. This section of the paper explains the process and workflow for repurposing metadata.

The authors identified record sets for transformation from XML documents using XSLT. For experimental purposes, the authors selected the collection sets, bypassing consultation with subject selectors or stakeholders, as would be the case when working with a collection identified for digitization and ingestion into the library's institutional repository. If a record set from the live catalog were being transformed, the collection would

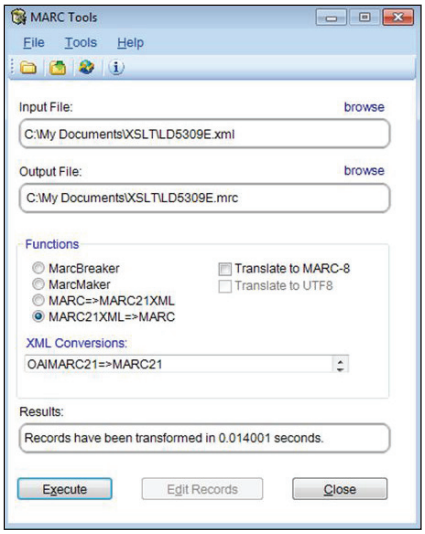

Figure 1. MARC21XML to MARC Conversion

have been first identified by a selector, cleared by the copyright librarian for inclusion in the institutional repository, and then ingested into the repository. As this was an experiment, none of the preparatory steps were necessary.

It is important to note that Texas A&M's local ILS is a relational database; therefore Microsoft Access was an excellent tool for use in querying the database. After identifying a collection set, an Access report was run to pull data from the local Voyager ILS. Initial queries involved gathering the bibliographic record ID numbers using two tables and limiting by series or call number in a third table. Depending on the collection, one can also run a Binary Large Objects (BLOB) query to search for information in any field of a bibliographic record. By querying the data, the authors obtained OCLC numbers from the 035 subfield a, in records identified in the initial queries as belonging to a collection set and fed them through the OCLC batch processing utility to obtain the desired MARC files from the OCLC database. The authors worked with OCLC records for this specific collection because those records might

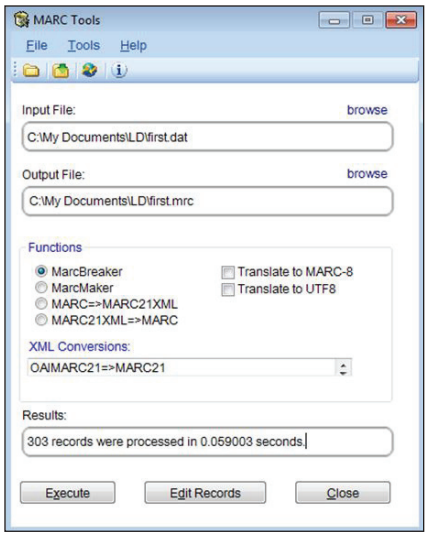

Figure 2. MarcBreaker Function

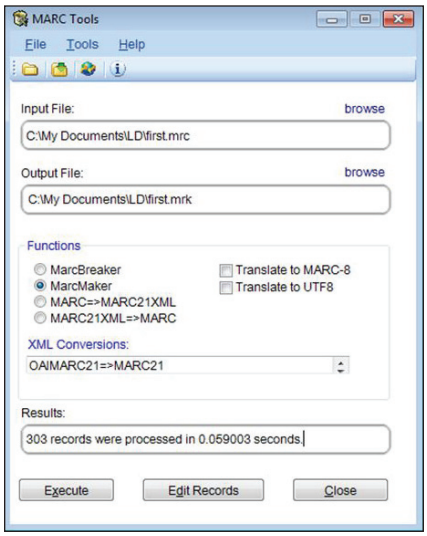

Figure 3. MarcMaker Function

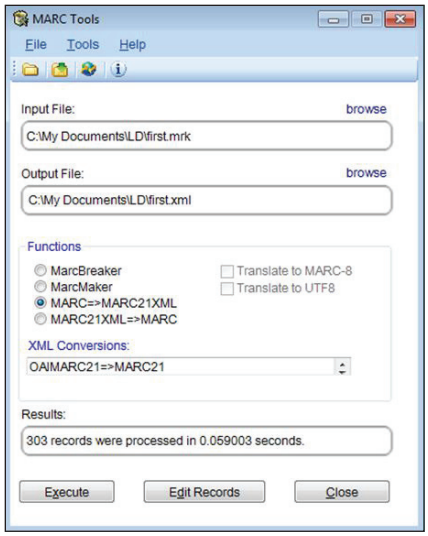

Figure 4. Marc to MARC21XML Function

have been updated or enhanced since first imported into the local ILS, thus potentially conceivably providing richer metadata.

After preparing the MARC files, MarcEdit was used to convert the files into MARCXML documents. The first step was to convert the downloaded OCLC file to MRC file format using the "MarcBreaker" function (see figure 2). The second step created the MARC file in MRK file format with the "MarcMaker" function (see figure 3). The final MarcEdit step was the transformation of the MRK file into XML format with the "MARC=>MARC21XML" function (see figure 4). Once the files were in XML format, the authors experimented with them by applying different stylesheets and transforming them from MARCXML to DC XML files. They used Oxygen software for processing and transforming the metadata.

Each collection had specific requirements, and to meet those requirements and preserve and transform metadata from MARC to DC, the authors tailored stylesheets for each collection. They first created a collection profile to match the local DSpace DC metadata matrix. The next step was the addition of fields that would be the same for every item in a collection.

Table 1. Sample Metadata Profile

MARC field	Dublin Core field
100, 110, 111	dc.creator
245	dc.title
260 subfield c	dc.date.created
260 subfield b	dc.publisher
650 _4 (example: Major Mathematics)	dc.description
500	dc.description
520	dc.description.abstract
Added MARC field 546	dc.language.iso
600	dc.subject.lcsh
650 _4 (example: Major Mathematics)	dc.description
830	dc:relation.isPartOfSeries
In some cases added field (594) with value: "Texas A&M University"	dc.publisher
Added field (590) with value: "text"	dc.type.material
Added field (595) with value: "reformatted digital"	dc.format.digitalOrigin
Added field (596) with value: "electronic"	dc.format.medium

These additional global fields included the following: dc:format.digitalOrigin, dc:format.medium, dc:type.material, dc:type, and dc:language. According to Hillmann, the recommended best practice for the values of the Language element is defined by RFC 3066 which, in conjunction with ISO 639, defines two and three letter primary language tags with optional subtags (see table 1).[37] The authors used the following form: en-US.

The command line for adding the language element in the prescribed format is:

```
<xsl:template match=
"marc:record">
<marc:record>
 <xsl:apply-templates/>
 <marc:datafield tag= "546">
 <marc:subfield code=
```

```
"a">en_US</marc:subfield>
 </marc:datafield>
```

Another example for adding a field that applies to all the records from the collection:

```
<marc:datafield tag="590">
 <marc:subfield code= "a">
 <xsl:text>text</xsl:text>
 </marc:subfield>
 </marc:datafield>
```

The authors selected 5XX fields that were initially omitted from the records to avoid duplication and loss of data.

The next step involved the creation of a stylesheet that mapped MARC data to DC data fields. This would be the actual crosswalk from MARCXML to DCXML. For all the trial collections, the 245 MARC title filed was mapped to dc:title, while the 100,110 and 111 field were mapped to dc:creator. Another example of mapping is in the publisher element, where the subfield b from the 260 MARC tag was mapped to dc.publisher. For some collection sets, it made more sense to add a separate field with the name of the institution mapped to dc:publisher element instead of mapping the subfield b from the 260 field to the publisher element. This enabled more uniform and consistent publisher names for all the records, particularly as the authors discovered that the OCLC records were not of a consistently good quality and sometimes needed enhancement. This process could only be used when the authors knew that the publisher was the same for all records in a collection. An example of a collection with the same publisher was the *Annual Budget Reports* for the Texas A&M University System units. In this case, the publisher for the entire collection is the Texas A&M University System. Catalogers also mapped specific subfield data to the DC element that appeared to be the best fit. An example of this is the 260 subfield c, which was mapped to dc:date.created. The command line for mapping the 260 subfield c to dc:date.created looks like this:

```
<xsl:for-each
select="marc:datafield[@
tag=260] ">
 <xsl:variable name= "date.
created">
 <xsl:call-template name=
"subfieldSelect">
 <xsl:with-param name=
"codes">c</xsl:with-param>
 </xsl:call-template>
 </xsl:variable>
 <dc:date.
created><xsl:value-of select=
"substring($date.created,
1, string-length($date.
created)-1) "/></dc:date.
created>
 </xsl:for-each>
```

If the collection had a series statement, the command for mapping that statement to dc:relation.isPartOfSeries element was added.

```
<xsl:for-each select=
"marc:datafield[@tag=830]">
 <xsl:variable name=
"relation.isPartOfSeries">
 <xsl:call-template name=
"subfieldSelect">
 <xsl:with-param name=
"codes">abcdevxyz034</
xsl:with-param>
 </xsl:call-template>
 </xsl:variable>
 <dc:relation.
isPartOfSeries><xsl:value-of
select= "substring($relation.
isPartOfSeries, 1,
string-length($relation.
isPartOfSeries)-1) "/></
dc:relation.isPartOfSeries>
 </xsl:for-each>
```

Every collection profile enumerated each specific metadata element necessary to describe the collection, and in essence, it served as a personal guide. Some collections benefited from having the 100 field mapped to dc:creator while the 7XX field was mapped to dc:contributor. In some cases that was not a good choice because the authors determined that individual authors would all be considered creators; in those cases the 100 field and all 7XX fields were mapped to dc:creator.

For some collections, the authors experimented with mapping the MARC 650 subfield z, the geographic subdivision, to dc:coverage:

```
<xsl:for-each select=
"marc:datafield[@tag=650]">
 <xsl:variable name=
"coverage">
 <xsl:call-template name=
"subfieldSelect">
 <xsl:with-param name=
"codes">z</xsl:with-param>
 </xsl:call-template>
 </xsl:variable>
 <dc.coverage><xsl:value-
of select=
"substring($coverage, 1,
string-length($place)-1)"/></
dc.coverage>
 </xsl:for-each>
```

This could not be applied to all collection sets, but for some, such as the *Annual Budget Reports* set, it was possible. Not all parts of every MARC record in a collection were mapped to the full DC schema in this experiment. For example, the linking field 776 (additional physical form) and other 7XX fields were not mapped to the dc:relation element.

The process of transforming MARCXML to DC records initially was a two-step process. The first step enhanced the MARCXML file with additional fields using one stylesheet and the next step transformed the enhanced MARCXML into DC using another stylesheet. Over the course of the project, the authors learned that they could combine these separate

steps into one process. They created a new stylesheet that provided both enhancement and transformation of the original MARCXML file (see appendix C). This stylesheet could be modified according to the needs of each collection.

The issue with crosswalking metadata from a schema with high granularity such as MARC to a less granular schema like DC is that the loss of bibliographic content and context is unavoidable. Reese notes that metadata of lower granularity cannot easily be moved to schemas with higher granularity because content and context cannot be manufactured if they are not present in the original record.[38] For this reason, the authors added fields to the MARCXML file that are equivalent across the whole collection by applying a transformation scenario through XSLT. Examples of added fields are the language field, the MARC 590 field with value "text" mapped to dc:type.material, or the MARC 591 field with a value "reformatted digital" mapped to datafield dc:format.digitalOrigin. This solution may not be perfect, but it was accomplished easily by using a stylesheet customized for each collection set. Careful analysis of content before the construction of a stylesheet for a collection enabled identification of fields that would be appropriate for all records. The number of addition global fields across a collection varied; up to ten were added for some collection sets.

The technical staff from the Digital Initiatives Department confirmed that the DC Resource Description Framework (DC-RDF) format is acceptable for ingestion of records into DSpace, Texas A&M University Libraries' institutional repository software. Initial experiments described in this paper proved to be successful, leading to inclusion of the Cataloging Department in metadata creation for the institutional repository.

One cataloger adjusted the stylesheets available on the LC's Network Development and MARC Standard Office site. The customization of the stylesheets for this experiment took five hours. Using this stylesheet, one thousand records were transformed from MARCXML format to DC XML format in less than a minute.

Discussion

Considering the fact that this work is duplicative within the institution and across the profession and within libraries at large, utilizing XSLT and similar programing languages plus tools that help with automating much of the processes is more than welcome. Automating these processes and the time saved by doing so cannot compare with the manual creation of either electronic equivalent of print records or equivalents in different metadata schema. As demonstrated in this paper, the authors created thousands of records within minutes using stylesheets that required five to ten hours to develop. The creation of a new basic record by cloning an existing record within an ILS can take several minutes, and using stylesheets to create thousands records can result in substantial savings in staff time. It should be noted that the authors are now re-using and adapting the same stylesheets, and therefore do not need to spend time creating new stylesheets each time. Simple modifications and adjustments to the existing ones take an insignificant amount of time.

Conclusion

Catalogers have the potential to undertake metadata projects by active participation in the transformation of MARCXML file format into DC XML or other metadata file formats. The authors, all catalogers, demonstrated that enhancement to MARCXML is possible with XSLT, and that creation of a MARCXML file format for electronic bibliographic records from print bibliographic records can be easily accomplished using XSLT stylesheets.

The authors also demonstrated that a crosswalk of records to another metadata schema, DC in this case, is simplified with XSLT. Catalogers' knowledge and understanding of metadata trends and various schemas should be utilized in the transformation and repurposing of existing metadata stored in MARC records. Library managers should encourage catalogers to learn programming languages and how to use free, open-source tools such as MarcEdit. They should also encourage similar projects that could reduce expensive data entry by transforming and reusing existing metadata. Essential skills for these transformations are an awareness of the importance of the use of established standards and of consistency and precision in data entry. Catalogers already possess those skills. With the inclusion of catalogers in metadata projects for institutional repositories, library administrators utilize valuable partners with requisite skill sets.

References

1. Jin Ma, "Metadata in ARL Libraries: A Survey of Metadata Practices," *Journal of Library Metadata* 9, no. 1–2 (2009): 1–14, 2013.
2. Jung-ran Park, Caimei Lu, and Linda Marion, "Cataloging Professionals in the Digital Environment: A Content Analysis of Job Descriptions," *Journal of the American Society for Information Science & Technology* 60, no. 4 (2009): 844–57, dx.doi.org/10.1002/asi.21007.
3. Ingrid Hsieh-Yee, "Educating Cataloging Professionals in a Changing Information Environment," *Journal of Education for Library & Information Science* 49, no. 2 (2008): 93–106, accessed May 3, 2013, www.jstor.org/stable/40323778.
4. Karen Calhoun, "Being a Librarian:

Metadata and Metadata Specialists in the Twenty-First Century," *Library Hi Tech* 25, no. 2 (2007): 174–87, dx.doi.org/10.1108/07378830710754947.
5. Terry Reese, "Automated Metadata Harvesting: Low-Barrier MARC Record Generation from OAI-PMH Repository Stores Using MarcEdit," *Library Resources & Technical Services* 53, no. 2 (2009): 121–34, dx.doi.org/10.5860/lrts.53n2.121.
6. Ibid.
7. Ibid.
8. Yuji Tosaka, "Analyzing Library Metadata for Web-Based Metadata Reuse Services: A Case-Study Examination of WorldCat.org and RefWorks," *Journal of Library Metadata* 10, no. 4 (2010): 257–75, dx.doi.org/10.1080/19386389.2010.524864.
9. Ibid.
10. Ibid.
11. Myung-Ja Han, "Creating Metadata for Digitized Books: Implementing XML and OAI-PMH in Cataloging Workflow," *Journal of Library Metadata* 11, no. 1 (2011):19–32, dx.doi.org/10.1080/19386389.2011.545001.
12. Mary S. Woodley et al., "Crosswalks, Metadata Harvesting, Federated Searching, Metasearching: Using Metadata to Connect Users and Information," in *Introduction to Metadata*, online edition, version 3.0 (Los Angeles: J. Paul Getty Trust, 2008), accessed July, 2013, http://scholarworks.csun.edu/bitstream/handle/10211.2/2001/WoodleyMary200803.pdf?sequence=1.
13. Margaret St. Pierre and William P. LaPlant Jr., "Issues in Crosswalking Content Metadata Standards," (white paper, National Information Standards Organization (NISO), Bethesda, MD, 1998), accessed May 3, 2013, www.niso.org/publications/white_papers/crosswalk.
14. Woodley, "Crosswalks."
15. Gordana Rudic and Dusan Surla, "Conversion of Bibliographic Records to MARC 21 Format," *Electronic Library* 27, no. 6 (2009): 950–67, dx.doi.org/10.1108/02640470911004057.
16. Naomi Dushay and Diane I. Hillman, "Analyzing Metadata for Effective Use and Re-use," in *Proceedings of the 2003 International Conference on Dublin Core and Metadata Applications: Supporting Communities of Discourse and Practice—Metadata Research & Applications* (n.p.: Dublin Core Metadata Initative, 2003) : 1-10, accessed July 30, 2013, http://dl.acm.org/citation.cfm?id=1383296.1383318.
17. Woodley, "Crosswalks."
18. St. Pierre and LaPlant, "Issues in Crosswalking Content Metadata Standards."
19. Carol Jean Godby, Jeffrey A. Young, and Eric Childress, "A Repository of Metadata Crosswalks," *D-Lib Magazine* 10, no. 12 (2004), accessed May 3, 2013, www.dlib.org/dlib/december04/godby/12godby.html.
20. Ibid.
21. Adam Chandler, and Elaine L. Westbrooks, "Distributing Non-MARC Metadata: The CUGIR Metadata Sharing Project," *Library Collections, Acquisitions & Technical Services* 26, no. 3 (2002): 207, dx.doi.og/10.1016/S14649055(02)00247-6.
22. Judith Ahronheim and Marko Lynn, "Exploding Out of the MARC Box: Building New Roles for Cataloging Departments," *Cataloging & Classification Quarterly* 30, no. 2–3 (2000): 216–25.
23. Woodley, "Crosswalks."
24. Calhoun, "Being a Librarian," 178.
25. Bruce Chr. Johnson, "XML and MARC: Which is 'Right'?" *Cataloging & Classification Quarterly* 32, no. 1 (2001): 81–90, dx.doi.org/10.1300/J104v32n01_07.
26. Ibid.
27. Corey Keith, "Using XSLT to Manipulate MARC Metadata," *Library Hi Tech* 22, no. 2 (2004): 122–30.
28. Jeni Tennison, *Beginning XSLT 2.0: From Novice to Professional* (Berkeley, CA : Apress, 2005), 50.
29. Ibid.
30. Ibid.
31. W3C, "XML Path Language (XPath) 2.0 (Second Edition)," accessed April 11, 2014, www.w3.org/TR/xpath.
32. Library of Congress, Network Development and MARC Standards Office, "MARCXML," accessed July 30, 2013, www.loc.gov/standards/marcxml.
33. Keith, "Using XSLT to Manipulate MARC Metadata."
34. Ibid.
35. Karen Coyle, "Understanding Metadata and Its Purpose," *Journal of Academic Librarianship* 31, no. 2 (2005): 160–63.
36. Library of Congress, Network Development and MARC Standards Office, "MARCXML."
37. Diane Hillmann, "Using Dublin Core—The Elements," Dublin Core Metadata Initiative, November 2005, accessed August, 3, 2013, http://dublincore.org/documents/usageguide/elements.shtml.
38. Terry Reese, "Automated Metadata Harvesting."

Appendix A. XSLT for Creation of Bibliographic Records for Electronic Information Resources from Bibliographic Records for the Print Version of the Same Resource

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:marc="http://www.loc.gov/MARC21/slim"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <xsl:output method="xml" indent="yes"/>
 <xsl:strip-space elements="*" />

 <!-- Output all elements verbatim -->
 <xsl:template match="*">
  <xsl:for-each select="marc:record"/>
  <xsl:element name="{name()}" namespace="http://www.loc.gov/MARC21/slim">
   <xsl:apply-templates select="@*"/>
   <xsl:apply-templates/>
  </xsl:element>
 </xsl:template>

 <!-- and all attributes -->
 <xsl:template match="@*">
  <xsl:attribute name="{name()}">
   <xsl:value-of select="."/>
  </xsl:attribute>
 </xsl:template>


 <!--
  remove a field
 -->
 <xsl:template match="marc:controlfield[@tag='001']"/>
 <xsl:template match="marc:controlfield[@tag='003']"/>
 <xsl:template match="marc:controlfield[@tag='005']"/>
 <xsl:template match="marc:datafield[@tag='029']"/>
 <xsl:template match="marc:datafield[@tag='035']"/>
 <xsl:template match="marc:datafield[@tag='300']"/>
 <xsl:template match="marc:datafield[@tag='776']"/>
 <xsl:template match="marc:datafield[@tag='891']"/>
 <xsl:template match="marc:datafield[@tag='850']"/>


 <xsl:template match="marc:record">
  <marc:record>
   <xsl:apply-templates select="marc:leader"/>
   <xsl:call-template name="t006"/>
   <xsl:call-template name="t007"/>
   <xsl:call-template name="t008"/>
   <xsl:apply-templates select="marc:controlfield[@tag &gt; '008']"/>
   <xsl:apply-templates select="marc:datafield[@tag &lt; '245']"/>
   <xsl:apply-templates select="marc:datafield[@tag='245']"/>
   <xsl:apply-templates select="marc:datafield[(@tag &gt; '245') and (@tag &lt; '300')]"/>
   <xsl:call-template name="t300"/>
```

```
    <xsl:apply-templates select="marc:datafield[(@tag &gt; '300') and (@tag &lt; '588')]"/>
    <xsl:call-template name="t588"/>
    <xsl:apply-templates select="marc:datafield[(@tag &gt; '588') and (@tag &lt; '700')]"/>
    <xsl:call-template name="t655"/>
    <xsl:apply-templates select="marc:datafield[(@tag &gt; '655') and (@tag &lt; '856')]"/>
    <xsl:call-template name="t856"/>
    <xsl:apply-templates select="marc:datafield[@tag &gt; '856']"/>
  </marc:record>
 </xsl:template>

 <xsl:template name="t006">
  <marc:controlfield tag="006">
   <marc:subfield code="a">m    d</marc:subfield>
  </marc:controlfield>
 </xsl:template>

 <xsl:template name="t007">
  <marc:controlfield tag="007">
   <marc:subfield code="a">c</marc:subfield>
   <marc:subfield code="b">r</marc:subfield>
  </marc:controlfield>
 </xsl:template>

 <xsl:template name="t008">
  <xsl:variable name="f008" select="concat(substring(marc:controlfield[@tag='008'],1,23),'o',substring(marc:controlfield[@
tag='008'],25))"/>
  <marc:controlfield tag="008">
   <xsl:value-of select="$f008"/>
  </marc:controlfield>
 </xsl:template>

 <xsl:template match="marc:datafield[@tag='245']">
  <xsl:variable name="i1" select="@ind1"/>
  <xsl:variable name="i2" select="@ind2"/>
  <xsl:variable name="f245a" select="marc:subfield[@code='a']"/>
  <xsl:variable name="f245a_endpunc" select="substring($f245a,string-length($f245a))"/>
  <xsl:variable name="f245b" select="marc:subfield[@code='b']"/>
  <xsl:variable name="f245c" select="marc:subfield[@code='c']"/>
  <marc:datafield tag="245" ind1="{$i1}" ind2="{$i2}">

   <xsl:choose>
    <xsl:when test="$f245a_endpunc = '.'">
     <marc:subfield code='a'>
      <xsl:value-of select="substring($f245a,1,string-length($f245a)-1)"/>
     </marc:subfield>
     <marc:subfield code='h'>
      <xsl:text>[electronic resource]</xsl:text>
      <xsl:value-of select="$f245a_endpunc"/>
     </marc:subfield>
    </xsl:when>

    <xsl:when test="$f245a_endpunc = '/' or $f245a_endpunc = ':'">
     <marc:subfield code='a'>
```

```
        <xsl:value-of select="substring($f245a,1,string-length($f245a)-2)"/>
       </marc:subfield>

       <marc:subfield code='h'>
        <xsl:value-of select="concat('[electronic resource]', ' ', $f245a_endpunc)"/>
       </marc:subfield>
      </xsl:when>
    </xsl:choose>
    <xsl:if test="$f245b">
     <marc:subfield code='b'>
      <xsl:value-of select="$f245b"/>
     </marc:subfield>
    </xsl:if>
    <xsl:if test="$f245c">
     <marc:subfield code='c'>
      <xsl:value-of select="$f245c"/>
     </marc:subfield>
    </xsl:if>
  </marc:datafield>
</xsl:template>

<xsl:template name="t300">
  <marc:datafield tag="300" ind1="" ind2="">
    <marc:subfield code="a">
      <xsl:text>1 online resource</xsl:text>
    </marc:subfield>
  </marc:datafield>
</xsl:template>

<xsl:template name="t588">
  <marc:datafield tag="588" ind1="" ind2="">
    <marc:subfield code="a">
      <xsl:text>Description based on print record.</xsl:text>
    </marc:subfield>
  </marc:datafield>
</xsl:template>

<xsl:template name="t655">
  <marc:datafield tag="655" ind1="" ind2="7">
    <marc:subfield code="a">
      <xsl:text>Electronic books.</xsl:text>
    </marc:subfield>
    <marc:subfield code="2">
      <xsl:text>local</xsl:text>
    </marc:subfield>
  </marc:datafield>
</xsl:template>

<xsl:template name="t856">
  <marc:datafield tag="856" ind1="4" ind2="0">
    <marc:subfield code="u">
      <xsl:text>http://digital.library.tamu.edu/digital-collections/texasfarmer/texas-farmer.html</xsl:text>
    </marc:subfield>
```

```
  </marc:datafield>
 </xsl:template>

</xsl:stylesheet>
```

Appendix B. Transformation Made to Bibliographic Records for Print Versions of Information Resources to Bibliographic Records for Electronic Information Resources

Record 1

```
</marc:record>
<marc:record>
<marc:leader>01238cas a22003377a 4500</marc:leader>
<marc:controlfield tag="001">ocm14108790 </marc:controlfield>
<marc:controlfield tag="003">OCoLC</marc:controlfield>
<marc:controlfield tag="005">20120503040056.0</marc:controlfield>
<marc:controlfield tag="008">860819c18939999txudx ne   0  a0eng c</marc:controlfield>
<marc:datafield tag="010" ind1=" " ind2=" ">
<marc:subfield code="a">sn 86088544 </marc:subfield>
</marc:datafield>
<marc:datafield tag="040" ind1=" " ind2=" ">
<marc:subfield code="a">PPM</marc:subfield>
<marc:subfield code="c">PPM</marc:subfield>
<marc:subfield code="d">NSD</marc:subfield>
<marc:subfield code="d">OCLCQ</marc:subfield>
</marc:datafield>
<marc:datafield tag="022" ind1="1" ind2=" ">
<marc:subfield code="a">1055-4726</marc:subfield>
<marc:subfield code="l">1055-4726</marc:subfield>
<marc:subfield code="2">1</marc:subfield>
</marc:datafield>
<marc:datafield tag="032" ind1=" " ind2=" ">
<marc:subfield code="a">045360</marc:subfield>
<marc:subfield code="b">USPS</marc:subfield>
</marc:datafield>
<marc:datafield tag="035" ind1=" " ind2=" ">
<marc:subfield code="a">(OCoLC)14108790</marc:subfield>
</marc:datafield>
<marc:datafield tag="042" ind1=" " ind2=" ">
<marc:subfield code="a">pcc</marc:subfield>
<marc:subfield code="a">nsdp</marc:subfield>
</marc:datafield>
<marc:datafield tag="049" ind1=" " ind2=" ">
<marc:subfield code="a">TXAM</marc:subfield>
</marc:datafield>
<marc:datafield tag="130" ind1="0" ind2=" ">
<marc:subfield code="a">Battalion (College Station, Tex. : 1893)</marc:subfield>
</marc:datafield>
<marc:datafield tag="222" ind1=" " ind2="4">
<marc:subfield code="a">The Battalion</marc:subfield>
<marc:subfield code="b">(College Station, Tex. 1893)</marc:subfield>
</marc:datafield>
<marc:datafield tag="245" ind1="0" ind2="4">
```

<marc:subfield code="a">The battalion.</marc:subfield>
</marc:datafield>
<marc:datafield tag="246" ind1="1" ind2="0">
<marc:subfield code="a">Texas A&M battalion</marc:subfield>
</marc:datafield>
<marc:datafield tag="260" ind1=" " ind2=" ">
<marc:subfield code="a">College Station, Tex. :</marc:subfield>
<marc:subfield code="b">The Austin and Calliopean Literary Societies,</marc:subfield>
<marc:subfield code="c">1893-</marc:subfield>
</marc:datafield>
<marc:datafield tag="265" ind1=" " ind2=" ">
<marc:subfield code="a">The Battalion, 230 Reed McDonald, Texas A&M University, College Station, TX 77843</
marc:subfield>
</marc:datafield>
<marc:datafield tag="300" ind1=" " ind2=" ">
<marc:subfield code="a">v. :</marc:subfield>
<marc:subfield code="b">ill ;</marc:subfield>
<marc:subfield code="c">40 cm.</marc:subfield>
</marc:datafield>
<marc:datafield tag="310" ind1=" " ind2=" ">
<marc:subfield code="a">Daily (weekdays except holidays, exam periods; Tues.-Fri. in summer),</marc:subfield>
<marc:subfield code="b"><Jan. 25, 1991-></marc:subfield>
</marc:datafield>
<marc:datafield tag="321" ind1=" " ind2=" ">
<marc:subfield code="a">Monthly,</marc:subfield>
<marc:subfield code="b">1893-19</marc:subfield>
</marc:datafield>
<marc:datafield tag="362" ind1="0" ind2=" ">
<marc:subfield code="a">Vol. 1, no. 1 (Oct. 1, 1893)-</marc:subfield>
</marc:datafield>
<marc:datafield tag="500" ind1=" " ind2=" ">
<marc:subfield code="a">At head of title, <Jan. 25, 1991->: Texas A&M.</marc:subfield>
</marc:datafield>
<marc:datafield tag="752" ind1=" " ind2=" ">
<marc:subfield code="a">United States</marc:subfield>
<marc:subfield code="b">Texas</marc:subfield>
<marc:subfield code="c">Brazos</marc:subfield>
<marc:subfield code="d">College Station.</marc:subfield>
</marc:datafield>
<marc:datafield tag="780" ind1="0" ind2="0">
<marc:subfield code="t">College journal</marc:subfield>
<marc:subfield code="w">(DLC)sn 88083304</marc:subfield>
<marc:subfield code="w">(OCoLC)17497658</marc:subfield>
</marc:datafield>
<marc:datafield tag="936" ind1=" " ind2=" ">
<marc:subfield code="a">Vol. 90, no. 80 (Friday, Jan. 25, 1991) LIC</marc:subfield>
</marc:datafield>
<marc:datafield tag="994" ind1=" " ind2=" ">
<marc:subfield code="a">C0</marc:subfield>
<marc:subfield code="b">TXA</marc:subfield>
</marc:datafield>
</marc:record>

Record 2

```
<marc:record>
  <marc:leader>01238cas a22003377a 4500</marc:leader>
  <marc:controlfield tag="006">
    <marc:subfield code="a">m    d</marc:subfield>
  </marc:controlfield>
  <marc:controlfield tag="007">
    <marc:subfield code="a">c</marc:subfield>
    <marc:subfield code="b">r</marc:subfield>
  </marc:controlfield>
  <marc:controlfield tag="008">860819c18939999txudx neo   0  a0eng c</marc:controlfield>
  <marc:datafield tag="010" ind1=" " ind2=" ">
    <marc:subfield code="a">sn 86088544 </marc:subfield>
  </marc:datafield>
  <marc:datafield tag="040" ind1=" " ind2=" ">
    <marc:subfield code="a">PPM</marc:subfield>
    <marc:subfield code="c">PPM</marc:subfield>
    <marc:subfield code="d">NSD</marc:subfield>
    <marc:subfield code="d">OCLCQ</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="022" ind1="1" ind2=" ">
    <marc:subfield code="a">1055-4726</marc:subfield>
    <marc:subfield code="l">1055-4726</marc:subfield>
    <marc:subfield code="2">1</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="032" ind1=" " ind2=" ">
    <marc:subfield code="a">045360</marc:subfield>
    <marc:subfield code="b">USPS</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="042" ind1=" " ind2=" ">
    <marc:subfield code="a">pcc</marc:subfield>
    <marc:subfield code="a">nsdp</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="049" ind1=" " ind2=" ">
    <marc:subfield code="a">TXAM</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="130" ind1="0" ind2=" ">
    <marc:subfield code="a">Battalion (College Station, Tex. : 1893)</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="222" ind1=" " ind2="4">
    <marc:subfield code="a">The Battalion</marc:subfield>
    <marc:subfield code="b">(College Station, Tex. 1893)</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="245" ind1="0" ind2="4">
    <marc:subfield code="a">The battalion</marc:subfield>
    <marc:subfield code="h">[electronic resource].</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="246" ind1="1" ind2="0">
    <marc:subfield code="a">Texas A&amp;amp;M battalion</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="260" ind1=" " ind2=" ">
    <marc:subfield code="a">College Station, Tex. :</marc:subfield>
    <marc:subfield code="b">The Austin and Calliopean Literary Societies,</marc:subfield>
```

```
    <marc:subfield code="c">1893-</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="265" ind1=" " ind2=" ">
    <marc:subfield code="a">The Battalion, 230 Reed McDonald, Texas A&amp;amp;M University, College Station, TX 77843</
marc:subfield>
  </marc:datafield>
  <marc:datafield tag="300" ind1="" ind2="">
    <marc:subfield code="a">1 online resource</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="310" ind1=" " ind2=" ">
    <marc:subfield code="a">Daily (weekdays except holidays, exam periods; Tues.-Fri. in summer),</marc:subfield>
    <marc:subfield code="b">&amp;lt;Jan. 25, 1991-&amp;gt;</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="321" ind1=" " ind2=" ">
    <marc:subfield code="a">Monthly,</marc:subfield>
    <marc:subfield code="b">1893-19</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="362" ind1="0" ind2=" ">
    <marc:subfield code="a">Vol. 1, no. 1 (Oct. 1, 1893)-</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="500" ind1=" " ind2=" ">
    <marc:subfield code="a">At head of title, &amp;lt;Jan. 25, 1991-&amp;gt;: Texas A&amp;amp;M.</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="588" ind1="" ind2="">
    <marc:subfield code="a">Description based on print record.</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="655" ind1="" ind2="7">
    <marc:subfield code="a">Electronic books.</marc:subfield>
    <marc:subfield code="2">local</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="752" ind1=" " ind2=" ">
    <marc:subfield code="a">United States</marc:subfield>
    <marc:subfield code="b">Texas</marc:subfield>
    <marc:subfield code="c">Brazos</marc:subfield>
    <marc:subfield code="d">College Station.</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="780" ind1="0" ind2="0">
    <marc:subfield code="t">College journal</marc:subfield>
    <marc:subfield code="w">(DLC)sn 88083304</marc:subfield>
    <marc:subfield code="w">(OCoLC)17497658</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="856" ind1="4" ind2="0">
    <marc:subfield code="u">http://digital.library.tamu.edu/digital-collections/texasfarmer/texas-farmer.html</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="936" ind1=" " ind2=" ">
    <marc:subfield code="a">Vol. 90, no. 80 (Friday, Jan. 25, 1991) LIC</marc:subfield>
  </marc:datafield>
  <marc:datafield tag="994" ind1=" " ind2=" ">
    <marc:subfield code="a">C0</marc:subfield>
    <marc:subfield code="b">TXA</marc:subfield>
  </marc:datafield>
 </marc:record>
```

Appendix C. Stylesheet to enhance and transform original MARCXML file

```
<xsl:stylesheet xmlns:marc="http://www.loc.gov/MARC21/slim"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 version="2.0" exclude-result-prefixes="marc">
 <xsl:import href="MARC21slimUtils.xsl"/>
 <xsl:output method="xml" indent="yes"/>

 <xsl:template match="/">
  <rdf:RDF>
   <xsl:apply-templates/>
  </rdf:RDF>
 </xsl:template>
 <xsl:template match="marc:record">
  <xsl:variable name="leader" select="marc:leader"/>
  <xsl:variable name="leader6" select="substring($leader,7,1)"/>
  <xsl:variable name="leader7" select="substring($leader,8,1)"/>
  <xsl:variable name="controlField008" select="marc:controlfield[@tag=008]"/>


  <rdf:Description>
   <xsl:for-each
    select="marc:datafield[@tag=120]">
    <dc:id>
     <xsl:value-of select="."/>
    </dc:id>
   </xsl:for-each>

   <xsl:for-each select="marc:datafield[@tag=245]">
    <xsl:variable name="title">
     <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abfghk</xsl:with-param>
     </xsl:call-template>
    </xsl:variable>
    <dc:title><xsl:value-of select="substring($title,1,string-length($title)-1)"/></dc:title>
   </xsl:for-each>

   <xsl:for-each
    select="marc:datafield[@tag=246]">
    <dc:title.alternative>
     <xsl:value-of select="."/>
    </dc:title.alternative>
   </xsl:for-each>

   <xsl:for-each
    select="marc:datafield[@tag=110]|marc:datafield[@tag=111]">
    <dc:creator>
     <xsl:value-of select="."/>
    </dc:creator>
   </xsl:for-each>

   <xsl:for-each
    select="marc:datafield[@tag=700]|marc:datafield[@tag=711]|marc:datafield[@tag=720]">
    <dc:contributor>
     <xsl:value-of select="."/>
```

```
  </dc:contributor>
</xsl:for-each>

<dc:type>
  <xsl:if test="$leader7='c'">
    <xsl:text>collection</xsl:text>

  </xsl:if>
  <xsl:if test="$leader6='d' or $leader6='f' or $leader6='p' or $leader6='t'">

    <xsl:text>student project</xsl:text>
  </xsl:if>
  <xsl:choose>
    <xsl:when test="$leader6='a' or $leader6='t'">text</xsl:when>
    <xsl:when test="$leader6='e' or $leader6='f'">cartographic</xsl:when>
    <xsl:when test="$leader6='c' or $leader6='d'">notated music</xsl:when>
    <xsl:when test="$leader6='i' or $leader6='j'">sound recording</xsl:when>
    <xsl:when test="$leader6='k'">still image</xsl:when>
    <xsl:when test="$leader6='g'">moving image</xsl:when>
    <xsl:when test="$leader6='r'">three dimensional object</xsl:when>
    <xsl:when test="$leader6='m'">software, multimedia</xsl:when>
    <xsl:when test="$leader6='p'">mixed material</xsl:when>
  </xsl:choose>
</dc:type>
<xsl:for-each select="marc:datafield[@tag=655]">
  <dc:type>
    <xsl:value-of select="."/>
  </dc:type>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=100]">
  <xsl:variable name="creator">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:creator><xsl:value-of select="substring($creator,1,string-length($creator)-1)"/>
  </dc:creator>
</xsl:for-each>


<xsl:for-each select="marc:datafield[@tag=260]">
  <xsl:variable name="date.created">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">c</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:date.created><xsl:value-of select="substring($date.created,1,string-length($date.created)-1)"/></dc:date.created>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=300]">
  <dc.description>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">ab</xsl:with-param>
    </xsl:call-template>
```

```
      </dc.description>
    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=500]">
      <xsl:variable name="description">
        <xsl:analyze-string select="marc:subfield[@code='a']" regex="(\w*)\s*process">
          <xsl:matching-substring>Printing process for original image: <xsl:value-of select="regex-group(1)"/> process.</
xsl:matching-substring>
          <xsl:non-matching-substring><xsl:value-of select="."/></xsl:non-matching-substring>
        </xsl:analyze-string>
      </xsl:variable>
      <dc:description><xsl:value-of select="substring($description,1,string-length($description)-1)"/></dc:description>
    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=502]">
      <dc.type.genre>
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">a</xsl:with-param>
        </xsl:call-template>
      </dc.type.genre>
    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=504]">
      <xsl:variable name="description">
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">a</xsl:with-param>
        </xsl:call-template>
      </xsl:variable>
      <dc:description><xsl:value-of select="substring($description,1,string-length($description)-1)"/></dc:description>

    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=515]">
      <xsl:variable name="description">
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">a</xsl:with-param>
        </xsl:call-template>
      </xsl:variable>
      <dc:description><xsl:value-of select="substring($description,1,string-length($description)-1)"/></dc:description>

    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=520]">
      <xsl:variable name="description.abstract">
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">a</xsl:with-param>
        </xsl:call-template>
      </xsl:variable>
      <dc:description.abstract><xsl:value-of select="substring($description.abstract,1,string-length($description.abstract)-1)"/></
dc:description.abstract>

    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=546]">
      <dc:language.iso>
        <xsl:call-template name="subfieldSelect">
```

```
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:language.iso>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=588]">
  <xsl:variable name="description">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:description><xsl:value-of select="substring($description,1,string-length($description)-1)"/></dc:description>

</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=590]">
  <dc:type.material>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:type.material>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=591]">
  <dc:type.genre>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:type.genre>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=594]">
  <dc:publisher>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:publisher>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=595]">
  <dc:format.digitalOrigin>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:format.digitalOrigin>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=596]">
  <dc:format.medium>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">a</xsl:with-param>
    </xsl:call-template>
  </dc:format.medium>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=600]">
```

```
  <xsl:variable name="subject.lcsh">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdefghjklmnopqrstuvxyz0234</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:subject.lcsh><xsl:value-of select="substring($subject.lcsh,1,string-length($subject.lcsh)-1)"/></dc:subject.lcsh>
 </xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=610]">
  <xsl:variable name="subject.lcsh">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdefghklmnoprstuvxyz0234</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:subject.lcsh><xsl:value-of select="substring($subject.lcsh,1,string-length($subject.lcsh)-1)"/></dc:subject.lcsh>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=611]">
  <xsl:variable name="subject.lcsh">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">acdefghjklnpqstuvxyz0234</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:subject.lcsh><xsl:value-of select="substring($subject.lcsh,1,string-length($subject.lcsh)-1)"/></dc:subject.lcsh>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=630]">
  <dc:subject>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdq</xsl:with-param>
    </xsl:call-template>
  </dc:subject>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=650]">
  <xsl:variable name="subject.lcsh">
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdevxy034</xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <dc:subject.lcsh><xsl:value-of select="substring($subject.lcsh,1,string-length($subject.lcsh)-1)"/></dc:subject.lcsh>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=660]">
  <dc:subject>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdevxy034</xsl:with-param>
    </xsl:call-template>
  </dc:subject>
</xsl:for-each>

<xsl:for-each select="marc:datafield[@tag=653]">
  <dc:subject>
    <xsl:call-template name="subfieldSelect">
      <xsl:with-param name="codes">abcdq</xsl:with-param>
    </xsl:call-template>
```

```
      </dc:subject>
    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=662]">
      <dc:coverage>
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">abcdefgh</xsl:with-param>
        </xsl:call-template>
      </dc:coverage>
    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=752]">
      <dc:coverage>
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">adcdfgh</xsl:with-param>
        </xsl:call-template>
      </dc:coverage>
    </xsl:for-each>


    <xsl:for-each
      select="marc:datafield[@tag=760]|marc:datafield[@tag=762]|marc:datafield[@tag=765]|marc:datafield[@
tag=767]|marc:datafield[@tag=770]|marc:datafield[@tag=772]|marc:datafield[@tag=773]|marc:datafield[@
tag=774]|marc:datafield[@tag=775]|marc:datafield[@tag=776]|marc:datafield[@tag=777]|marc:datafield[@
tag=780]|marc:datafield[@tag=785]|marc:datafield[@tag=786]|marc:datafield[@tag=787]">
      <xsl:variable name="relation">
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">ot</xsl:with-param>
        </xsl:call-template>
      </xsl:variable>
      <dc:relation><xsl:value-of select="substring($relation,1,string-length($relation)-1)"/></dc:relation>

    </xsl:for-each>

    <xsl:for-each select="marc:datafield[@tag=830]">
      <xsl:variable name="relation.isPartOfSeries">
        <xsl:call-template name="subfieldSelect">
          <xsl:with-param name="codes">abcdevxyz034</xsl:with-param>
        </xsl:call-template>
      </xsl:variable>
      <dc:relation.isPartOfSeries><xsl:value-of select="substring($relation.isPartOfSeries,1,string-length($relation.
isPartOfSeries)-1)"/></dc:relation.isPartOfSeries>
    </xsl:for-each>

    <dc:language.iso>en_US</dc:language.iso>
    <dc:type.material>text</dc:type.material>
    <dc:publisher>Texas A&amp;amp;M University</dc:publisher>
    <dc:format.digitalOrigin>reformatted digital</dc:format.digitalOrigin>
    <dc:format.medium>electronic</dc:format.medium>

  </rdf:Description>
 </xsl:template>
</xsl:stylesheet>
```