2005-12-20

MARC vs XML

Peter A. Zuber
peter_zuber@byu.edu

MARC vs XML

December 20, 2005

Peter A. Zuber

Background

MARC (MAchine-Readable Cataloging record) is considered “the most important development in the history of library automation.”[1] MARC defines a formatting standard by which a bibliographic record that includes (among other items) descriptions, titles, subject headings and classification or call number can be formatted in such a way as to be automatically read and interpreted by a particular type of machine (computer).

The MARC development was headed by Henriette Avram of the Library of Congress during the mid 1960’s and evolved into a suite of related standards during the 1980’s (USMARC, Can/MARC, InterMARC, UKMARC, etc.) that defined the MARC concept for participating nations. During the 1990’s the standard was again revised with the intent of harmonizing a consistent standard for participating nations and is called MARC 21.[2] As can be seen, this revision activity was driven not only by a need to further refine and improve MARC but also driven by the inherent advantages that existed within the MARC concept. That is, a cataloging record standardized in its format, standardized in its content and made machine-readable became extensible in its application. In this state, a bibliographic record could be exchanged and shared between library sites, enabling it to reach far beyond the local library. No longer were patrons and libraries limited to internal collections in the card catalog but could extend their search to any participating library that used and followed the MARC standard. The very nature of this “control and standardize” effort coupled with computer prowess created a more open and accessible library environment. Notably, this effort, first launched over forty-five years ago, has had such a lasting and profound impact on the state of the industry. But that long-term success leaves some librarians feeling unease about the prospects of changing to another system.

As of late, MARC is under a great deal of scrutiny, primarily in what it’s perceived it can’t do. With the advent of the web and its growth as a provider of rich content and media, many have seen the forty-five year old standard as inadequate to the task, lacking some of the fundamental tools required to deliver such content to users no longer satisfied with just concise bibliographic information.

During the course of this paper, some preliminary explanation of the nature of MARC and XML, regarded as the next most likely step beyond MARC, will be provided as well as a narrative on the issues arising between each format and potential gains and losses in their use.

The MARC Record

A MARC record consists of a listing of bibliographic information formatted according to the rules defined by the MARC 21 standard.[3] For example, bibliographic records are typically divided into fields such as author, title, publisher, etc. In MARC, these fields are realized as “tags” and assigned a specific three-digit number depending on the nature of the field. For example, the ISBN (International Standard Book Number) is the bibliographic field assigned to the “020” tag. When a 020 tag appears, the ISBN will follow. Within a tag, it is possible to define subfields that hold specific bibliographic data within the field. For example, the “100” tag is assigned to the personal name main entry. Since names are given in various forms, such as family, surname, forename, subfield codes are used to tell computers which form is used. In the example below, the

100 entry for Martin Luther King designates three subfields, “a” for the name, “c” for titles or other words associated with a name and “d,” dates associated with a name.

100 1_ $a King, Martin Luther, $c Jr., $d 1929-1968.

This is a typical representation of a MARC field. The tag’s structure provides a means to add attributes to the field that may or may not be considered hierarchal in nature. The “1_” between the tag and subfields is the indicator which, depending on the tag itself, can carry different meanings. In this case, “1_” means the name in subfield “a” is given in single surname form. The dependence in this case of the indicators to the tag also extends to the subfields. Subfield codes, although they may have the same alpha descriptor, mean very different things depending on the tag being used.

The intent of this brief exercise was not so much to teach MARC but to show some of the more interesting attributes of MARC that should draw attention. The first, as suggested by Yee, is how “MARC 21 is a data structure standard, not a data content standard or a data value standard. A data structure standard provides a standard for the labeling of data and, as such, for the isolation of particular kinds of data for particular purposes such as indexing or display. The data itself (or the semantic content), however, is determined by data content standards (cataloging rules such as AACR2R) and data value standards (lists of authorized headings, such as the National Name Authority File or LCSH).”[4] Notable in her description is the statement that this concept “seems to be misunderstood by some,”[5] suggesting the general understanding of the role MARC fulfills in the cataloging process may be somewhat exaggerated. According to William

Lund, Director of Library and Information Science at Brigham Young University, “where the actual content in MARC is very tightly controlled by different standards, it is possible to create a valid MARC record without using AACR2, LCSH, etc. MARC determines the format of the data and what information is required, but the standards determine how to represent that information.”[6] The importance of content standards that are effectively independent of encoding formats cannot be understated.

Secondly, an extremely important aspect of MARC is the dependence of indicator and subfield code definitions on the particular tag being used. This tightly controlled system effectively requires the use of extensive documentation in order to ensure the right tag is used as well as the correct indicators and subfields within that tag. Where this aspect provides strict control over record creation, it does present a very challenging task to the novice or marginally experienced cataloguer to correctly create MARC records. It has been derisively said that the only “people who believe themselves able to read a MARC record without referring to a stack of manuals is a handful of top cataloguers and those on serious drugs.”[7]

The XML Record

The World Wide Web Consortium published XML, or Extensible Markup Language, in December 1997.[8] XML was created through the evolutionary process of web development, basically a symbiosis of the SGML standard (or Standard Generalized Markup Language) and HTML (HyperText Markup Language). The most notable language from this standard is HTML, the standard web language (at least for the present) for building pages and including content. From the growing need to provide interoperability, migration and platform independence XML was born. Tennant describes HTML as being a language that "can't do much more than describe the look of a web page, whereas SGML is too complicated and unwieldy for most applications. XML achieves much of the power of SGML without the complexity and adds web capabilities beyond HTML."[9]

The growth of this language is evidenced in its use not only in web design but also as a tool in stand-alone applications for enabling connectivity between newer devices and older software. Its applications and inherent power are such that even HTML has been enhanced to XHTML, now growing in popularity as the new standard. It is in essence a tool for structuring data. It isn't a programming language and doesn't require a programmer to write or use it. It is license-free, extensible, platform-independent, and supports internationalization and localization. Being fully Unicode-compliant, it can use data in 16 rather than 8 bit chunks in order to correctly represent foreign characters.[10] Basically, it's free; it can be whatever you want it to be; and it can run on just about any computer system, anywhere. One of its more attractive aspects is that the XML file is text, plain and simple. Not intended to be read like a text document, it nonetheless reveals its pattern, methodology, and purpose in a simple viewing of the file. This makes it extremely easy not only to comprehend but to debug as well.

XML is used for structuring data and uses tags and attributes similar in form to HTML. However, unlike HTML, it uses tags to delimit pieces of data instead of specifying what each tag and attribute means. For example, in XML, the tag "<p>" can mean anything. In HTML, "<p>" means paragraph, and unless the HTML specification is changed, it will always mean paragraph. The advantage with XML is the developer can decide exactly what tags mean, how they should be structured and what they contain, and create as many as needed. It is a blank sheet, willing to be whatever the developer wants it to be, leaving the interpretation of its tags to the application that is accessing the XML record. XML does require that the structure be correctly formed. It won't be as accommodating as an HTML file, willing to run even if errors exist. In this effort, there are tools available that determine whether an XML document is well formed, meaning that the basic standards for using XML are followed, regardless of the content. In addition, there are tools to determine whether an XML document complies with a specific content standard. This is "good practice" methodology, which is exactly what the library industry should expect.

The use of an XML file does require other supporting files for the data to make sense. A XML style sheet called the XSLT, or Extensible Stylesheet Language Transformation, provides the function of transforming XML data into the form or forms desired. It can be considered much like a simple program that reads and uses the data as desired. If the XML information is to be displayed in a web browser, a HTML CSS, or Cascading Style Sheet, is used for that purpose.

The Question of Content

Some of the issues in discussion between MARC and XML deal with the desire to expand the content of the bibliographic record. An often-cited example is the library that wishes to enrich the current electronic record with a table of contents. Although a table of contents inclusion is actually possible with the MARC record, it is certainly not an ideal platform for such content. The MARC record is described as being a "flat" record, [11] meaning, as mentioned previously, it does not have a “hierarchal” nature. As such, it is a poor candidate for hierarchal structures such as a table of contents. Miller percieves that this flat structure, by nature, creates an “decided underemphasis on relationships.”[12] However, to this point, Yee suggests that there is a limit to “the degree of hierarchicality [sic] that can be supported in the current shared cataloging environment,”[13] meaning the record by virtue of its independence and ability to move in and out of different systems was designed this way to be cost efficient. Others argue that MARC is hierarchal to the extent of tags and subfields, and that the data content (again more related to “data content standards” and not MARC) is highly hierarchal. Regardless, no one argues the hierarchal nature of XML and its clear superiority, at least in this regard, to MARC.

In addition, other desirable content would be such media savvy items as book cover graphics, book jacket information or even reviews. Where some of this content might be included, it is difficult to see how MARC could provide a system for graphical display. Interestingly, the Functional Requirements of Bibliographic Records (FRBR), a report published by the International Federation of Library Associations and Institutions (IFLA), outlines “a revolutionary recasting of the bibliographic record on behalf of library users. It defined such principles as the hierarchical dimensions of a creative product: work (distinct creation), expression (realization of a work), manifestation (physical embodiment) and item (a single exemplar).”[14]

This effectively places all variations of a work into a single record and allows users the means to narrow in on the content of interest.

The Question of Granularity

Other issues focus on “poor granularity” of MARC records, meaning the amount of information provided is minimalist by nature. Where this may be the case in practice, it seems unreasonable to attribute this as characteristic of the MARC format. There are certainly provisions in the MARC standard for adding additional bibliographic information. It would seem the degree to which the record needs to be, should be, or wants to be filled is the question at hand. This is more of the cataloguer’s decision rather than a restriction of the format. Indeed, if the XML format was used, is there any guarantee that a higher granularity would be realized? An XML file could be as poor in granularity as a MARC record.

The Question of Multiple Scripts

If a bibliographic record having multiple scripts, for instance multi-lingual titles, is desired, multiple title fields can be tagged in MARC using 246 and 880 tags.[15] For example, the main title in Russian using Latin characters can be placed in the 246 tag, while the same title in Russian using Cyrillic characters can be placed in the 880 tag. This provides a nice way to include and link different forms of the language, yet does not allow an indication as to the form used. If a patron wanted to view the Russian title in Cyrillic, there is no indication of what field contains that data. In XML, tags can contain attributes such as “script = russian.latin” and “script = russian.cyrillic,” thus allowing the display of data to be in the format the user desires. This is in part due to the flexibility of XML, at least, what it COULD be in the bibliographical world.

Technical "Marginalization"

Certainly, the life span of MARC is impressive in any context. The fact that the standard lives in the world of computer language and is still useful after forty-five years seems incomprehensible. Languages such as COBOL, FORTRAN and PASCAL are either answers to trivia questions or distant memories today, yet they were actively used up until the late 1980's and early 1990's. It is a tribute to the MARC creators that it has set a standard of usefulness that likely will never be matched in the computer language industry. It is a clear example of a standard clearly purposed and correctly envisioned. It was born in a time when a computer filled a room, when memory, processing ability, CPU time and storage capability were at a premium. What the MARC standard did, along with computer usage, for libraries worldwide is inestimable. Having said this, there is concern expressed for this formatting standard's future in an age of sophisticated computer systems, massive memory and display capabilities, and demanding users that expect more in the "information age." The MARC standard, according to Lund, "is not very portable and probably only supported on library public access catalog systems."[16] Is the MARC 21 standard up to the challenge? While the rest of the world seemly embraces the XML standard in many different working environments and applications, the library industry is part early adopter, part hard-core traditionalist. Baruth suggests that "for many librarians the advantages of an XML based local library system seem vague and not worth the cost to change. There is the fear that XML will not survive, leading us down an endless migration path."[17] There are also emotional aspects clearly expressed in the "Libraryland Lexicography"[18] where "XML" stands for "Ex-Master Librarian." Another contributor may be the notion that valuable information in existing MARC records would be “lost in translation” to XML. This is certainly a legitimate concern, especially when other translation initiatives like the Dublin Core, which “offers only 15 elements and obviously cannot mimic the richness of MARC records with their hundreds of data elements”[19] is considered. But at least in this regard, XML has no limit to data elements it is willing to accept.

There is no argument that the XML standard allows for potentially greater gains in descriptive content and the ability to fully leverage the technical prowess that modern computer systems provide. It is to this point that the question on marginalization should reside. It is not marginalization if the intent is never to use such capabilities. The MARC standard, as McDonough offered, “exists only to enable bibliographic systems to exchange data. Interoperability between systems is already accomplished, and ‘XML-lizing’ [sic] MARC won’t do anything to make that particular application work any better.”[20]

Where many programs exist for taking MARC records and converting them to XML (most notable being the resources provided by the Library of Congress), [21] simple conversions prompt several questions, one of which was expressed by Needleman. “Taking the current incarnation of MARC and recasting it in XML is a mechanical exercise that doesn’t really accomplish very much. If we are going to recast MARC, what we really need to be doing is looking at.…what possibilities exist for doing things in new and better ways in XML that are not possible given the structure of the current MARC formats.”[22]

In fact, it seems this question of “marginalization” is not meaningful unless a willingness to redesign or launch a new standardization effort is accepted.

Loss or Gain?

The struggle is to determine if migration toward XML is worth the effort and to consider what would be gained and what would be lost. In doing such, it is instructive to review the progression of a successful computer language as a model for what could happen. During the early 1980’s the “C” computer language was gaining ground as a viable contender for tasks normally associated with other code types such as FORTRAN, COBOL and BASIC. There were several aspects of C which provided a great deal of flexibility and power while at the same time improving processing power and reducing CPU time, all of which were significant contributions in the days of lean computer platforms. The migration started slowly, aided by early adopter mentality and preliminary examples of efficient code at use. With the advent of C++, new ground was broken in terms of Class Structures and Object Oriented Programming that would make this variation the obvious choice over almost any language. Later enhancements such as Visual C++ provided a coding environment that produced more professional looking programs, faster turn around, and leveraged the power of the resident operating system. Even today the once simple “C” continues to migrate, with variations such as Visual C++ .NET (a more web centric platform) and C#. This evolution took no more than 15 years to reach its present state and, in fact, follows the personal computer’s growth almost identically. It is interesting to see in this industry, especially during the time its hardware was evolving from infancy, the rapid progress a simple programming language took to become an extremely sophisticated product line. It is sobering to imagine the short duration (as well as number of variations) XML might take today, especially when hardware growth does not stunt its progression.

One of the enablers of the rapid growth spurt of “C” was the lack of industry standardization. The latest iteration that showed promise was immediately adopted as the development standard, partly in fear of being left behind if the older product was retained, partly in response to increased capability. The effect was to generate enormous expenditures in company re-tooling, employee training (or more likely, employee lay off and rehire), as well as down time until the companies’ current shipping product was recast in the code *du jour*. This is an important consideration in the dilemma of migrating MARC to XML. XML is from the same “society” of developers that produced FORTRAN, C, C++, Visual C++ and so on. It is unlikely to expect that the practice of revision will stop, especially since a close cousin HTML, already embedded world-wide, is itself under constant revision.

The concern of migrating all content to XML while not knowing what XML will be in 10 years is legitimate. This requires addressing the question that was mentioned before. Why was MARC so successful over so many years? It seems, certainly in comparison to the “C” programming example, an essential component of MARC was its adherence to a strict specification and industry compliance to the standard. In addition, it combined both an encoding format and a structure that dictated content according to specification. Where XML is strictly an encoding format, it would seem reasonable that the next important steps in its evolution, at least as far as the library world is concerned, are a “data content standard” and a “data value standard” that address the issues mentioned previously. The Library of Congress, through its MARC to XML initiative provides support for the migration of MARC data into XML files. This puts XML in the same class as MARC, at least in so far as purpose is concerned. However, to leverage the power of XML, an effort similar to the one launched forty-five years ago is required, with the intent not just to redefine MARC in an XML world but to create an enhanced MARC that uses XML as its platform. Where the FRBR certainly appears viable in this regard, an enhanced standard will allow MARC to grow beyond its current specification by virtue of native XML capabilities. This enhanced version would, by definition, keep all the current MARC content and structure and then add to it, providing means to include graphics, web display, table of contents, interoperability outside of the library system, and so on. Again, because XML is freely licensed and “defined” by the XSLT, it can be whatever the library industry wants it to be for as long as it wants. However, where one of the advantages of XML is its ability to migrate to “outside” (meaning, non-library) systems, it is troubling to think that may be compromised over the years as the general computing industry, as it is prone to do, migrates to whatever is the latest and greatest. The saving aspect of XML in this regard is its open, definable nature, willing to be recast via the XSLT and CSS into what the current technology requires. That provides some safeguard that the enormous amount of data already coded in MARC format will continue to be viable and extensible in an XML environment for a long time to come.

It seems the popular plea “Why can’t we all just get along?” is apt in this case. XML is a very versatile tool, willing to be whatever it is designed to be and capable of providing significantly more content in a more broadly based environment than MARC could ever reach. The two do not have to be conflicting standards; in fact, arguably XML can be the vehicle “for transformations to and from other metadata approaches, such as Dublin Core and the Metadata Object Description Schema (MODS), helping to standardize derivative metadata records.”[23] XML can use the MARC structured concept to ease migration, all the while opening the door to more content and capability. XML simply provides more paper on which to write more, rather than erase what’s already there.

REFERENCES

1) Jaeso, P. (2002, September). XML and Digital Librarians. *Computers in Libraries*, pp. 46-49.

2) Library of Congress, Network Development and MARC Standards Office. (1998, October). *MARC 21: Harmonized USMARC and CAN/MARC*. Retrieved November 1, 2005, from http://www.loc.gov/marc/annmarc21.html

3) Library of Congress, Network Development and MARC Standards Office. (2005, September). *MARC Standards*. Retrieved November 2, 2005, from http://www.loc.gov/marc.

4) Yee, M. M. (2004). New Perspectives on the Shared Cataloging Environment and a MARC 21 Shopping List. *Library Resources & Technical Services*, *48* (3), 165-178.

5) Walch, V. I. (1994). *Standards for Archival Description: A Handbook*. Chicago: Society of American Archivists.

6) (W. Lund, personal communication, November 16, 2005).

7) Tennant, R. (2002, October 15). MARC Must Die. *Library Journal*, pp. 26-27.

8) World Wide Web Consortium, Architecture Domain. (1996). *Extensible Markup Language (XML)*. Retrieved November 1, 2005, from http://www.w3c.org/XML

9) Tennant, R. (2001, March 15). XML: The Digital Library Hammer. *Library Journal*, pp. 30-31.

10) World Wide Web Consortium, W3C Communications Team. (1999). *XML in 10 Points*. Retrieved November 1, 2005, from http://www.w3.org/XML/1999/XML-in-10-points

11) Tennant, R. MARC Must Die.

12) Miller, D. R. (2000). *XML and MARC: A Choice or Replacement?* Retrieved November 5, 2005 from Stanford University website: http://elane.stanford.edu/laneauth/ALAChicago2000.html

13) Yee, M. M. New Perspectives on the Shared Cataloging Environment and a MARC 21 Shopping List.

14) Tennant, R. (2003, April 15). Not Your Mother's Union Catalog. *Library Journal*, p. 28.

15) Lam, K.T. (2001, July). *Moving from MARC to XML - Part Two: Handling of Multi-Script Metadata.* Retrieved on November 1, 2005, from http://ihome.ust.hk/~lblkt/xml/marc2xml_3.html

16) (W. Lund, personal communication, November 16, 2005).

17) Baruth, B. (2002, June/July). Missing Pieces That Fill in the Academic Library Puzzle. *American Libraries*, pp. 58-63.

18) Schnider, K. G. (1998, June/July). Libraryland Lexicography, *American Libraries*, p.122.

19) Jaeso, P. XML and Digital Librarians.

20) McDonough, J. (2001, February 20). *Re:UNIMARC vs. XML*. Message posted to XML4LIB electronic mailing list, archived at http://lists.webjunction.org/wjlists/xml4lib/2001-February/004502.html

21) Library of Congress, MARC Standards. (2004, July). *MARC in XML*. Retrieved on November 1, 2005 from http://www.loc.gov/marc/marcxml.html

22) Needleman, M. (2001, February 19). *UNIMARC vs. XML*. Message posted to XML4LIB electronic mailing list, archived at http://lists.webjunction.org/wjlists/xml4lib/2001-February/004492.html

23) Library of Congress announces standard MARCXML Schema, *Online Libraries & Microcomputers*, June 2002, vol. 20, Issue 6/7. Retreived from http://web12.epnet.com.