
An introduction to the Metadata Object Description Schema (MODS)

Sally H. McCallum

The author

Sally H. McCallum is Chief of the Network Development and MARC Standards Office, Library of Congress, Washington, DC, USA.

Keywords

Online cataloguing, Extensible markup language

Abstract

This paper provides an introduction to the Metadata Object Description Schema (MODS), a MARC21 compatible XML schema for descriptive metadata. It explains the requirements that the schema targets and the special features that differentiate it from MARC, such as user-oriented tags, regrouped data elements, linking, recursion, and accommodations for electronic resources.

Electronic access

The Emerald Research Register for this journal is available at www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at www.emeraldinsight.com/0737-8831.htm

Publications, even print but especially electronic, are growing in number and catalogers must have viable options for speeding up the process of bibliographic control. The Metadata Object Description Schema (MODS), a MARC21 companion, is one of those options. MODS is intended to address the need for a MARC21 derivative that:

- takes advantage of the XML environment;
- gives special support to cataloging electronic resources;
- is less detailed; and
- most importantly, is highly compatible with MARC21.

MODS fits together with other XML schemata and tools that are available through the MARC Web site[1]. These include a MARC21 in XML (MARXML) that is completely round-trip-compatible with MARC21 in its ISO 2709 structure, i.e. the structure used currently for the format, along with transformations between the two (Corey Keith is reporting on MARXML in the next issue). Through MARXML, transformations among MODS, ONIX and Dublin Core are also provided at the site in addition to several tools.

An initial version of MODS was developed by the MARC21 maintenance agency at the Library of Congress with a group of MARC21 users. The draft (Version 1.0) was made widely available for review and trial use for six months in 2002 and a new version, that incorporated the changes suggested during review, was made available in February 2003 (Version 2.0). The MODS listserv has provided an essential avenue for user input and discussion of solutions. Based on comments received for the period February-August 2003, Version 3.0 was issued for review and comment in September 2003. The schema and accompanying documentation are available from the MODS Web site[2]. Various transformations to and from MARC21 and Dublin Core (DC) are also available from that site.

In this discussion, some key terms used by librarians will frequently be adapted to the new terminology of the environment in which MODS functions. A format will usually be

Received 17 September 2003

Revised 15 October 2003

Accepted 7 November 2003

© Sally H. McCallum.

called a “schema,” because that is the word used to describe the specification for a tag set and its rules for use in an XML setting. Cataloging data will sometimes be called descriptive metadata to relate them to and distinguish them from other types of “data about data”, such as technical, administrative, and rights, that play a prominent role with digital resources. Using this terminology, MODS is an XML schema for descriptive metadata.

Crafting a schema in XML presents a developer with many different options. The basic components of XML are elements and attributes associated with the elements, all of which behave according to a set of rules for XML established by the World Wide Web Consortium (W3C)[3]. XML is both the mark-up used for MODS records and a formal language for specifying the MODS schema. In the following discussion and examples of MODS mark-up these conventions will be used: MODS tag names will be enclosed in angle brackets (< >); attributes’ names and values will be in italic; data content elements will be in bold. Any of the above may sometimes be underlined to highlight them. While the angle brackets for tags are XML specifications, the italic, bold, and underlining are not.

Basic requirements for MODS

MODS was designed with the following basic requirements as guidelines.

XML environment

One fundamental requirement was that MODS be developed for eXtensible Mark-up Language (XML) environments. The library community is both looking for and accustomed to long-term solutions. Library automation began with the start of the commercial computer era, in the 1960s, and libraries have built a robust data interchange and multisystem vendor environment. This has served libraries’ advanced requirements (variable length data fields when the norm was fixed length, and extensive character encodings long before Unicode) but it required specialized software

and hardware to accommodate those requirements.

Although XML may appear to be a sudden development because of its rapid take-up, it is grounded in work of at least 20 years, starting with the ISO standard for the Standard Generalized Mark-up Language (SGML). SGML became familiar to many through the SGML-conforming tag set, HTML, the primary mark-up used for the early Web. XML followed SGML and HTML, drawing from experience with both, and was made more flexible yet consistent and predictable. It is maturing and appears to have good staying power. Thus it is a “methodology” into which the library environment can usefully move for new long-term solutions. Unicode is another positive aspect of the XML environment of which MODS takes advantage.

XML accommodates variable length data, explicit data tagging to multiple levels, hierarchical structure (even better than MARC), and “all possible” characters (through Unicode). It is likely that new-generation systems from the computer manufacturers will incorporate aspects of XML where useful and appropriate. Open source software is being built around the XML data structure, which makes manipulation of XML data easier (and cheaper), and enables what was formerly difficult. With XML, libraries would not always have to construct their own application tools. Therefore there was no question about using XML for MODS, just the options to be chosen for constructing the MODS schema.

Among the new XML tools of special interest to the library community are several protocols and broader metadata carriers. The search and retrieval protocol Z39.50, which is widely employed for search interoperability between dissimilar systems, can return several types of records. Although it is largely used for MARC21 and other non-XML record types, MODS-formatted records would be an option. However, the Search/Retrieve Web Service (SRW), the “next generation” Z39.50 which is built on the Z foundation, is an XML protocol and XML-formatted records are the natural record types for SRW[4].

The harvesting protocol of the Open Archives Initiative (OAI)[5] also transports metadata records and, being an XML protocol, XML

records are the best fit. The records that the Library of Congress currently makes available for OAI harvesting may be obtained in any of three XML formats: MODS, MARCXML, and DC.

An important use for MODS will be for the descriptive data in Metadata Encoding and Transmission Standard (METS) packages[6]. METS is an XML framework schema that carries all types of digital object metadata – structural, administrative, technical, and rights, in addition to descriptive. The metadata themselves may be part of the METS package or external, and XML is preferred. Either MODS or MARCXML records may be included, but one attractive possibility is for the METS package to contain the (sometimes) briefer MODS record and perhaps point to the fuller MARC21 or MARCXML record.

An additional reason for using XML is that electronic resources themselves are often XML, XML-compatible, or specially XML-receptive, so that metadata that are intended to accompany a resource are useful if encoded according to a recognized XML schema. While librarians would perhaps prefer a MARCXML record embedded in the header of a resource, MODS may be a more reasonable alternative, being simpler yet MARC-compatible.

Highly compatible with MARC21

MARC21 has been a long-term solution for the library community for a descriptive metadata format, and as a result there are over a billion MARC21 records in networks and local systems world-wide. There are thousands of MARC-based installed systems that carry out complex operations from end-user services to a library's processing needs. There are also thousands of MARC-trained librarians who have responsibility for the organization of library resources. The community must therefore provide a continuity and connectivity between the existing MARC21 records and any new XML record. They must be highly integrable, so existing systems can use them and bibliographic specialists can construct and understand them. The difference in the record structure (ISO 2709 vs XML) is not critical, but the compatibility of the semantics of the record content with the records already created is critical.

MODS was therefore designed as a MARC21 derivative, with clear recognition of the importance of supporting data constructed by the content rules currently used. For MODS it was also recognized that, while the primary target is compatibility with MARC data, MODS records should be able to derive data both from Dublin Core records which are extremely simple, and from publishers' ONIX records which are more complex and less compatible with library data.

Special accommodation of digital resources

Based on experience from several digital projects, there is a requirement for some special accommodations for electronic resources in MODS. This type of material is currently presenting the greatest challenge to librarians. The volume is enormous and growing, traditional selection procedures are not sufficient for selecting for inclusion to collections, "adding to a collection" is not even a well-understood concept for this type of material, and preservation is still an unknown. There is an added complexity in that the material is easily changed, so that even electronic versions of scholarly journals can introduce changes and corrections, and versioning for Web documents is especially daunting. All these issues pointed to special accommodation even from the descriptive metadata viewpoint.

Simplicity

With the growth of print and especially electronic resources, all descriptive metadata will not be equal. There will be resources that should have full description using the detail of a MARC record; others need simpler, perhaps even temporary, cataloging that can be partially derived from the resources or from other records such as ONIX or DC. As noted, all must be highly integrable with MARC21 records. That is a target that MODS has tried to hit by specifying only core data and essential tagging.

The top level of data elements in MODS has been kept modest, with 19 currently defined, although many of them have several sub-elements. A listing of those top elements indicates the completeness of coverage, however. Using the descriptive metadata

categories defined in the *Functional Requirements for Bibliographic Records (FRBR)* (IFLA Study Group on the Functional Requirements for Bibliographic Records, 1998) (hierarchical from the top down: work, expression, manifestation, and item), the top level MODS elements are:

Work: <titleInfo type="uniform">, <name>, <genre>, <targetAudience>, <classification>, <subject>

Expression: <typeOfResource>, <language>, <abstract>, <tableOfContents>

Manifestation: <titleInfo>, <originInfo>, <physicalDescription>, <note>, <identifier>, <relatedItem>

Item: <location>, <accessCondition>

The MODS elements <recordInfo> and <extension> are outside the basic FRBR framework.

An accompanying concern was that MODS should be simple enough for original description of a resource by non-professionals, with adequate guidance. This meant that, in addition to the relative simplicity of the schema, an accompanying document with use guidelines was needed (creation and use of that document and experience with MODS record generation are described in the paper by Rebecca Guenther in this issue).

Features of MODS

The following features of MODS illustrate how the above requirements were fulfilled.

User-oriented tags

For MODS the decision was made to use tagging that was more user-friendly in the sense that they can be read and generally understood by the uninitiated. While there is efficiency in brief tagging such as that used by MARC (three digit tags), their meaning must be learned or interpreted. The MODS tags are English language and few abbreviations are used. Examples are <title> for the MARC21 245 subfield a, <genre> for 008 genre information, and <publisher> for 260 subfield b. The following piece from a MODS record shows the <titleInfo> top-level tag and its subtags:

```
<titleInfo>
<title>Sound and fury :</title>
<subTitle>the making of the punditocracy
</subTitle>
</titleInfo>
```

Thought has been given to writing transformations that would convert the tags to and from other languages. For this to work, the tags would have to be standard in other languages and a strict one-to-one correspondence would be necessary, as they are critical for efficient information sharing.

Part of the intent with the use of word tags was to enable creation of records, with less training on aspects of the format and content designation. With the great volume of electronic resources, technicians will be employed for minimal level and initial bibliographic descriptions. In the university setting these are often university student assistants who come and go with the school years, and a schema with easy to understand tags is important.

Regrouped data elements

Some data elements that appear in various fields in MARC have been brought together in different ways in MODS. Placement in MARC sometimes reflects post-coordinated growth of the format for the various types of material, differences in cataloging rules over time, and data relationships as perceived at a certain time. While retaining basic correspondence to MARC, MODS has in some instances brought similar-coded values or data together, or separated data that are combined in MARC.

For example, MARC fields 440, 490, 534, 700-711 (when they contain subfield t), 730-740 (when indicator 2 has value 2), 760-787, and 800-830 all contain information about an item related to the resource being described. They come together under repeated occurrences of the MODS <relatedItem> tag, differentiated with tag attribute values that indicate the relationship. The following example shows two related items for an item, the series and a component part:

```
<relatedItem type="series">
<titleInfo>
<title>Music for voice and instrument
</title>
</titleInfo>
```

```

</relatedItem>
<relatedItem type="constituent">
<titleInfo>
<title>Tutto in pianto il cor struggete;
arr.1984.</title>
</titleInfo>
<name type="personal">
<namePart>Joseph I, Holy Roman Emperor,
</namePart>
<namePart type="date">1678-1711
</namePart>
</name>
</relatedItem>

```

Another interesting area where like information was usefully brought together was the MODS `<originInfo>`. Under this tag is place of publication information from MARC 008, 044, and 260 subfield a; publisher information from MARC 260 subfield b; date information from MARC 260 subfield c, 033, and 008; edition information from MARC 250; and issuance/frequency information from the MARC Leader and fields 310 and 321:

```

<originInfo>
<place>
<placeTerm type="code" authority
="marccountry">nyu</placeTerm>
<placeTerm type="text">Ithaca, N.Y
</placeTerm>
</place>
<publisher>Cornell University Press
</publisher>
<dateIssued>c1999</dateIssued>
<dateIssued encoding="marc">1999
</dateIssued>
<issuance>monographic</issuance>
</originInfo>

```

The MODS tag `<physicalDescription>` brings together a number of pieces of related information about a resource. Here, form of item information in MARC 008 is combined with the data in fields 256, 300, and part of field 856. A physical description might look like the following:

```

<physicalDescription>
<form authority="marcform">print
</form>
<extent>1 score (12 p.) + 2 parts ; 31 cm.
</extent>
</physicalDescription>

```

Another place where this occurred is with genre, which is primarily in coded data fields in MARC21. MARC information in various 007/01, 008/21, 008/24, 008/25, 008/26, 008/26, 008/30-31, and 008/33 bytes and in the 665 field are combined under one tag, `<genre>`.

Fewer coded values

For data elements that have widely established coded values, MODS accommodates specification of the code, but also provides for the text form of the element. An example is `<place>` under `<originInfo>` above, where both nyu and Ithaca, NY are recorded. In other cases where there are not universal lists, word value lists have been established. In the `<physicalDescription>` example above, under `<form>` the value print is used. This comes from a list established in coordination with the MARC codes values for the form of an item, as found in 008/23 and 008/29.

Electronic resource data

Since the schema was drafted with electronic resources as a primary target, several small but important elements were included that do not have good equivalents in MARC. Under `<physicalDescription>`, one is an indication of `<digitalOrigin>`, with values born-digital and formatted-digital; and another is the `<reformattingQuality>`, with values access, preservation, and replacement. Provision is made for the date that a Web site was captured as part of harvesting activities and also for the date of last access for remote Web resources. MODS also provides for differentiating between the URL-like strings used as identifiers, and those that are "actionable." The former are in the `<identifier>` tag, while the latter are under `<location>`.

Linking

MODS takes advantage of the linking flexibility of the XML environment by providing for all the top-level elements with attributes such as xlink and ID. The attribute xlink allows the encoding of a relevant link for the information, similar to links MARC allows in selected fields. MARC abstract (520) and table of contents (505) fields both have subfields for URLs that

link to relevant content. The attribute ID allows for linking internally.

Recursion

MODS takes advantage of the hierarchical elegance of XML to define the structure for related items in a recursive manner. Each related item may be described with the same tags as the resource targeted by the record, although typically only a small subset is needed. The example above for a `<relatedItem>` shows `<titleInfo>` and `<name>` elements being used recursively under the top level `<relatedItem>` element.

Special attributes

Some XML attributes occur in several MODS elements. Obvious ones give the authority for the data content, e.g. for classification, whether the class information is ddc, lcc, or some other classification authority. Another gives information on the encoding pattern for an element like a date. But the more interesting attributes are the following four:

- (1) lang;
- (2) xml:lang;
- (3) script; and
- (4) transliteration.

MODS defines them for all top level elements. MARC does not provide for identification of these attributes at the field level, because there are issues associated with their use and problems with coding techniques. While these problems were recognized, the decision was made to include them to see if they could be useful in actual practice and perhaps some issues settled.

Round-trip transformation with MARC21

Since MODS contains tagging for a subset of MARC21 and also has a few elements not contained in MARC, round-trip conversion without loss may not be possible in some instances. When this is important for a process or application, users could limit elements to those with one-to-one correspondences. In many cases the loss is slight, but data can always be preserved by using local tagging in MARC or the `<extension>` element in MODS.

XML tags

Two basic XML structure-related decisions involved the tagging rules to be used in the schema: mixing of content within a tag, and use of tags alone or tags and attributes.

Mixed content

In early drafts, a number of MODS elements were defined as “mixed-content”. This mixing of sub-elements with content is allowed if the schema declares that it is to be used. For example, a mixed-content approach was considered for the MODS `<title>` element, to indicate that a portion of a title is an initial article:

```
<title><nonSort>The</nonSort>world of
learning</title>
```

However, XML developers are reluctant to use the mixed-content approach because of the possibility of ambiguity and difficulty in referencing the content for processing, so MODS eventually decided against it. Thus, in the above example, `<title>` was enveloped in `<titleInfo>` and individual elements were each tagged:

```
<titleInfo><nonSort>The
</nonSort><title>world of learning</title>
</titleInfo>
```

Attributes vs elements

A decision had to be made whether to use attributes and elements in the schema or just elements. There is the question of whether attributes complicate processing of searches, displays, or transformations. No clear answer to that question has emerged, since there is no real evidence that one or the other approach is better. Searching depends on the query language, and the development of XML query languages is still not mature.

Another issue with attributes related to conversion of a MODS record to a format suitable for transmission via a Web protocol such as SOAP. This is an important issue, because there is some thought that an XML attribute cannot be encoded as a SOAP parameter. However, a MODS record would be transmitted as a string, not with parameters visible to SOAP.

In the final analysis, with no concrete evidence against the use of attributes, MODS

defines attributes wherever they are a useful construct. The general rule used is: if the information is “content”, it is made an element; if the information is used to aid in the processing of content, it is treated as an attribute. There are also two cases, however, where information must be cast as an element and not as an attribute:

- (1) if the information is repeatable; and
- (2) if it is structured (since XML elements may have sub-elements but attributes may not).

Conclusion

MODS is viewed as providing an evolutionary pathway forward for libraries. It attempts to take into account the rapid increase in electronic resources, the community’s economically-deep commitment to MARC data elements, the proliferation of formats and schemata beyond library community control, and the rapidly growing XML tool environment. MODS derives from MARC21 while taking advantage of XML and the flexibility, tool development, and transformation options it offers.

Notes

- 1 The MARC21 Web site contains format information including the MARCXML tools and schemata. It links to the MODS site, and is available at: www.loc.gov/marc
- 2 The MODS Web site contains information on the development of the schema, user guidelines, downloadable transformations, and the schema itself, available at: www.loc.gov/mods
- 3 On the World Wide Web Consortium Web site, available at: www.w3.org/XML/
- 4 For information on Z39.50 and links to the SRW Web site, available at: <http://lcweb.loc.gov/z3950/agency/>
- 5 Available at: www.openarchives.org/
- 6 Available at: www.loc.gov/mets/

Reference

- IFLA Study Group on the Functional Requirements for Bibliographic Records (1998), *Functional Requirements for Bibliographic Records: Final Report*, K.G. Saur, München, UBCIM Publications. New Series; 19, also available on IFLANET at: www.ifla.org/VII/s13/frbr/frbr.pdf or www.ifla.org/VII/s13/frbr/frbr.htm