

Chapter 7 Supplement

MODS: The Metadata Object Description Schema

Questions for Review, Study, or Discussion

1. What are some reasons why learning about MODS can be valuable even to those who do not and will not ever use it in practice?
2. What is MODS? Who uses MODS?
3. How is MODS structured? What are “container” elements and how do they differ from other elements in MODS? What are subelements? Can subelements be container elements? In what ways does the MODS scheme make use of XML element attributes?
4. How does the MODS scheme allow for flexibility in level of detail and granularity of the metadata itself? What are some examples?
5. What are the twenty top-level MODS elements? Which of these are container elements that include subelements? Which are non container elements and have no subelements?
6. Most of the examples used to illustrate various MODS elements, subelements, and attributes in Section 7.2 (pages 169-195) are the same as those used to illustrate Qualified Dublin Core in Chapters 3-4. These examples are also given side-by-side for comparison in a supplementary document on the book’s companion website. See also Table 7.2 on pages 205-207 of the text, which is a less detailed “Side-by-Side Comparison of Qualified Dublin Core and MODS Records.” What similarities and differences do you see between these two metadata schemes, their elements, and their structure?
7. What differences in potential user functionality do you see when comparing the different MODS and Qualified Dublin Core elements? How is MODS able to address some common resource description challenges in ways the DC is not?
8. What MODS elements, attributes, and attribute values are designed to explicitly deal with digitized versions of originally analog resources?
9. What is the purpose of the MODS *KeyDate* attribute?
10. How does the MODS *RelatedItem* element compare to the Dublin Core *Relation* and *Source* elements? In what ways can this element, at least theoretically and potentially, fulfill the One-to-One Principle at the same time as including individual metadata properties and values about both an original and a digital format of the same resource in a single MODS record?
11. What kinds of information does the MODS *RecordInfo* element contain, and why and to whom might these be useful?
12. What are some differences between the oXygen XML Editor and the University of Alberta MODS Editor in the way that the metadata creator enters and views MODS metadata in each?
13. How are MODS records displayed and transformed?
14. What is “mapping” in the metadata context? What is “MODS Lite?”
 - a. What are some differences between automated and human mapping between simple DC and MODS Lite, and between Qualified Dublin Core and full MODS?
 - b. What are the limitations inherent in automated mapping and why is this so? Why does automated mapping need to use the “defaults” given in the Notes column of the Simple Dublin Core Mapping to MODS shown in Table 7.3 on page 210?
 - c. How might different levels of complexity and sophistication in automated mapping potentially address some of these issues? Where are computers still likely to be unable to make intelligent

mapping decisions, at least at their current level of development, and as affordable by most cultural heritage institutions?

Recommended Readings and Resources for Reference or Further Study

Official MODS Technical Documents

- MODS: Metadata Object Description Schema Official Web Site. Library of Congress. <http://www.loc.gov/standards/mods/>
- “MODS User Guidelines (Version 3).” Library of Congress. Last modified October 27, 2010. <http://www.loc.gov/standards/mods/userguide/>.
- “Outline of Elements and Attributes in MODS Version 3.4.” Library of Congress. Last updated October 18, 2010. <http://www.loc.gov/standards/mods/mods-outline.html>.
- MODS “Guidance.” Library of Congress. Last modified January 11, 2011. <http://www.loc.gov/standards/mods/mods-guidance.html>
 - Sources
 - Value Lists
- MODS “Conversions.” Library of Congress. Last modified December 2, 2010. <http://www.loc.gov/standards/mods/mods-conversions.html>.
 - Includes “Dublin Core (simple) to MODS” and “MODS to Dublin Core (simple)” mappings.

Other MODS Documents

- DLF Aquifer Metadata Working Group. 2009. “Digital Library Federation / Aquifer Implementation Guidelines for Shareable MODS Records.” Version 1.1. March. https://wiki.dlib.indiana.edu/confluence/download/attachments/24288/DLFMODS_ImplementationGuidelines.pdf.
- Riley, Jenn. 2009. DLF Aquifer MODS Guidelines Levels of Adoption. Edited June 30. <https://wiki.dlib.indiana.edu/display/DLFAquifer/MODS+Guidelines+Levels+of+Adoption>

Readings

- Riley, Jenn, John Chapman, Sarah Shreeves, Laura Akerman, and William Landis. 2008. “Promoting Shareability: Metadata Activities of the DLF Aquifer Initiative.” *Journal of Library Metadata* 8, no. 3: 221–248.
- Guenther, Rebecca S. 2003. “MODS: The Metadata Object Description Schema.” *Libraries and the Academy* 3, no.1 (January): 137–150.
- Guenther, Rebecca S.. 2004. “Using the Metadata Object Description Schema (MODS) for Resource Description: Guidelines and Applications.” *Library Hi Tech* 22, no. 2: 89–98.

Exercises

Recommended Exercises

1. Take an existing Dublin Core metadata record and map the *Simple Dublin Core to MODS Lite* following the DC-to-MODS mapping table in Table 7.3 on page 210 of the book or on the MODS website. Also use the *Conversion of DC Resource Type Vocabulary to MODS TypeOfResource Values* in Table 7.4. or on the MODS website.

- a. Distinguish between what could be accomplished by means of machine mapping versus human mapping.
 - b. Then try mapping Qualified Dublin Core to a richer level of MODS, but still restrict yourself to what a computer could do by default.
 - c. Then compare what additional mapping aspects could be accomplished by a human being versus a machine.
2. Working closely with the full *MODS User Guidelines* online, create one or more original MODS records. Use one or more real or hypothetical digital resources, especially those that could make use of the full richness and a variety of different MODS elements and attributes. There is great value for learning both XML structures in general, and the MODS scheme and structure specifically, by typing out the entire record by hand, using line breaks and indentations. Alternatively, or in addition, an XML editor, such as the ones listed in the Chapter 6 exercises, could be used to create an original MODS record.
 3. Refer back to Chapters 3 and 4, and compare how Dublin Core and MODS are able to handle common resource description and user information retrieval needs.
 - a. See also the side-by-side comparison of *Dublin Core and MODS Element Description Examples* taken from Chapters 3, 4, and 7, available in the second section of the book's companion website.

Suggestions for Instructors

- Provide students with one or more records for mapping and be prepared with your own answers, which may well include some instance or more than one valid mapping.
- Provide students with a resource and/or resource information for describing with a complete MODS record. Or have students create a complete MODS record for one of the resources used for the Chapters 3-4 exercises, but this time without any reference to Dublin Core, taking MODS on its own terms.
- Discuss differences in functionality between Simple Dublin Core, Qualified Dublin Core, and MODS Lite, and full MODS.
- Consider creating a MODS XML record with a few intended errors in element nesting, use of attributes or the like, and have students find the errors. If using XML editing software, have the student try validating the erroneous MODS record against the MODS XML Schema, and find and fix the errors so that the file validates.