

An investigation of METS as a method of packaging metadata and data

Document details

Project: SHERPA DP
Work Package: 4.2 & 5.8
Author: Michael Popham
Version: Version 1.0
Document date: 28/04/06
Change history:

<i>Date</i>	<i>Version</i>	<i>Author</i>
28/04/06	First version	Michael Popham

Contents

An investigation of possible uses of METS to package e-prints metadata	2
The METS digital library metadata framework	2
METS, endorsed external schemas, and application profiles	2
METS and metadata for e-prints.....	3
An Assessment of METS as a mechanism for transferring data from DSpace and EPrints repositories	4
METS' provision for the transmission of (meta)data	4
Practical considerations for the transmission of (meta)data.....	4
References:	6

An investigation of possible uses of METS to package e-prints metadata

The METS digital library metadata framework

The METS XML Schema (currently version 1.5 [1, 5]) provides a metadata framework for digital library objects that implements a modular approach to metadata. In this modular approach, each metadata type (descriptive, administrative, structural etc.) can contain any number of distinct METS metadata sections. For example, descriptive metadata can be contained in any number of descriptive metadata sections (<dmdSec>). Usually, specialised external schemas, which serve a more narrowly defined purpose, are employed within these sections. This makes it possible to have e.g. a descriptive metadata section (<dmdSec>), which contains a DC-based record primarily intended for resource discovery via OAI-PMH, in parallel with a MODS [2]-based <dmdSec>, intended for resource description and interoperability within a hybrid library context, or indeed a <dmdSec> that employs a schema (application profile) specifically designed for a particular resource type, e.g. e-prints.

The METS Schema offers several metadata sections for this purpose (all of which are optional and repeatable): the <dmdSec>s for descriptive metadata records, the <amdSec>s for administrative metadata records, which are subdivided into technical metadata <techMD>, rights metadata <rightsMD>, metadata about the original <sourceMD>, and digital provenance metadata <digiprovMD>, and the structural map <structMap>, which records the internal, hierarchical structure of a digital object. The <structMap> is a prerequisite for the object's representation(s) and is the only compulsory element in a METS document. The container-based approach of the METS framework is flexible, and descriptive rather than prescriptive in the use of the metadata schemes employed. This modularity makes METS an ideal candidate to serve as an Information Package (IP) in the OAIS Reference Model [3]. There have already been concrete developments in this area that have focussed e.g., on the "Content Information" (CI) and "Preservation Description Information" (PDI) containers of the OAIS' A IP [9], and on the use of METS for SIP- and DIP-documents in institutional repository software [10, 13].

METS, endorsed external schemas, and application profiles

The METS framework implements an "object-oriented" approach. A METS document generally represents one digital object, e.g. an e-print, which can contain many parts, and which holds and/or references all the metadata and data the digital object is comprised of. A METS document can also contain specific <behavior>-definitions associated with the object's contents that record information on how to process or render it. The distinct sections of the METS document are held together by a system of interlinked XML IDs and IDREFs throughout. In order to keep METS documents more standardised and more easily exchangeable with other institutions working in similar areas of application, the METS editorial board has made recommendations on the metadata schemes used within METS. These include: DC, MODS, and MARC-XML for descriptive metadata, and TEXTMD, MIX, METSRights [1], IMD/AMD/VMD, and DIGIPROVMD [11] for administrative metadata. The PREMIS [12] preservation metadata implementation is now also available as a number of external XML schemas. It can be used within METS on its own using the PREMIS container schema and referencing it from an <amdSec>, and also as part of an application profile specifically designed for the preservation of a particular resource type, in which case individual schemas can be used in the four containers provided by the <amdSec>. As PREMIS does not focus on descriptive or technical metadata for images, it will complement the schemas already recommended for METS, particularly in its <digiprovMD>, <rightsMD>, and <techMD> sections. Most of the "standard" extenders are defined as external schemas, and as they have been designed by specialists in a particular domain, they will dictate greater standardization of vocabularies used and encoding schemes (cataloguing rules) applied in each element than any custom-built application profile that takes into account institutional and/or community-driven developments. The effort of mapping existing data may therefore be considerable. For new data (if not already in place), quality-assurance in the form of editorial

oversight would probably be essential to maintain the quality of metadata and validity of documents across all collaborating institutions.

METS and metadata for e-prints

Applied to e-prints in an institutional repository setting, the use of METS relies on either the modularization of existing metadata elements used across all project partners participating in SHERPA-DP and the application of one or more of the standardized external schemas to one or more of the METS sections, or the development of an application profile for e-prints that can be defined as a specialised extender of the METS Schema for one or more sections. While the former allows for greater standardization independent of the resource types held across all institutional repositories, the latter is more flexible and could take into account non-standards based, localized, or community-driven developments. In any case, the adopted use of METS should be defined in a METS profile, which is intended for public reference among project partners and external interested parties. This could aid software developers in adapting applications to handle METS documents, e.g. enable OAI-PMH harvesters to process a METS document containing or referencing an OAI-based record (as for example envisioned for version 1.4 of DSpace [10]), unless these are already provided to them through XSLT, for example.

It should be possible to accommodate a majority of SHERPA-DP descriptive metadata fields in standard extenders, such as MODS and PREMIS, although this may require a domain-specific extension to these standards. Some of the existing administrative metadata can be accommodated in the <metsHdr> if it concerns administration of the created METS document itself, and some in the <amdSec> containers if it relates to the content of the METS document. Any such decision requires strategic consideration and decisions, and possibly contractual agreements among the project partners involved. It is therefore outside the scope of this work package to attempt a mapping of the existing SHERPA-DP metadata to one or more of the METS Board-approved external schemas or indeed to suggest an application profile for SHERPA e-prints. Either way is possible and could be considered under strategic, architectural, and pragmatic points of view.

An Assessment of METS as a mechanism for transferring data from DSpace and EPrints repositories

METS' provision for the transmission of (meta)data

As its name implies, METS is intended for encoding, packaging, and transferring metadata in a networked environment. In accordance with the object-oriented approach taken by METS, the METS XML Schema also provides for a mechanism to embed base64-encoded arbitrary binary data in METS documents. The METS wrapper element defined for this purpose is `<binData>`, the equivalent of the element `<xmlData>`, which is intended for XML-encoded data. These elements can be contained both within the `<mdWrap>` element (for XML-encoded or base64-encoded metadata) and within the `<FContent>` element (for XML-encoded or base64-encoded data). If the metadata or data to be included is already available or encoded easily in XML then `<xmlData>` is a reasonable approach, if it is not and may be too costly or time-consuming to do, but would still be interesting to associate and distribute with the item, then it could just be included in `<binData>` and maybe taken care of later on in the workflow.

The `<binData>` element itself is defined as the XSD binary datatype "base64Binary", which must contain a finite-length, ordered sequence of octets in accordance with the base64 algorithm defined in RFC 2045 [6] and the characters defined in XML 1.0 (Second Edition) [4] as white space. Structurally, the `<FContent>` element, which would be used to transport base64-encoded data in a METS document, is contained within the `<file>` element and thus part of `<fileSec>`, the section of a METS document in which the individual files that comprise the digital object are either referenced or included. It is on this level of the `<file>` element that additional information about the data embedded can be captured, such as its MIME type, its size, a timestamp, and the type and value of the checksum mechanism used. The encoding of the content as base64 text is implicit in the element itself.

Practical considerations for the transmission of (meta)data

Since the METS community itself has not explored METS's mechanism for the transmission of base64-encoded data in METS documents in any great detail, the following considerations are the result of preliminary investigations into the potential issues involved, and are not based on insights gained through real-life implementations. It might be useful to explore how other communities have used this mechanism in their XML applications. The architectural, technical, and workflow requirements for the exchange of METS documents between institutional repositories have been explored in detail by several institutions, recently for example in the "Repository Bridge: Automated Linkage of National and Institutional Repositories" project [7], which implements the exchange between DSpace and FEDORA repositories based on a METS/OAI solution. In this case the data itself is only referenced for subsequent file transfer, however, and not embedded in the METS documents transferred. All requirements regarding system security, checksums, provisions for versioning, error handling and disaster recovery outlined there will apply to the exchange of METS documents holding base64-encoded data.

Initial tests have shown that base64-en/decoding is quick enough for on-the-fly implementation in the projected workflow of data exchange between repositories, but the real overhead implications of these additional steps can only be assessed in a distributed test environment, which is closely modelled on the architecture of the envisioned final system. Given that the container-based structure offered by the METS framework can be relatively easily created and populated automatically, it should be possible to implement a mechanism into EPrints or other software to create a skeletal METS record from the input provided by the depositor. It is likely that this initial METS document will need quality checking and editing, but it could serve as a starting point (SIP) for a workflow that is XML/METS-based from the start and will consequently require only expertise in XML editing, which should already be readily available at the project partner institutions involved. There are a number of tools and utilities

available on the METS home page [1], which help with processing METS documents, that might be useful for implementers of such a workflow.

One consideration is that the size of the resulting METS document could make it difficult to handle if subsequent events in the lifecycle of the document require changes in the METS file. Base64-encoding of binary content increases the file size by approximately a third assuming UTF-8 is the underlying encoding. This would be relevant if the use of METS were intended to serve as an OAIS AIP [see WP 4.2 a.], as AIPs are living documents that will require updating according to events in the course of their preservation. As common applications for handling XML documents tend to use the DOM model (Document Object Model), which requires the entire XML tree to be available in memory, this can become both laborious and resource-intensive. This is of course less of an issue if the METS document serves the purpose of an OAIS SIP or DIP only and is created (automatically) for the sole purpose of data exchange. It is also likely that the institutional repository software used will support some of these steps.

An advantage is that the actual digital content can be protected e.g. by a digital signature issued by the original content creator and/or subsequent distributors of the object, thus ensuring its authenticity. This is not the case when data is just referenced via URIs, in which case checksums can still be provided for each element referencing the content [8]. In either case, it will be essential to store a checksum in the METS document for each of the files listed in the <fileSec>, and ideally to do a check of the file against the checksum every time it is transmitted to ensure integrity of the data. In case it makes sense to pack and/or compress the binary data before inclusion into the METS document, the compression used and step(s) necessary to unpack the content need to be documented in the METS file as well. The current version of the METS Schema provides for the inclusion of such information in the <transformFile> element, which can be contained in the <file> element mentioned above. Since the latest revision of METS, the <file> element can be used recursively "to deal with [the] PREMIS onion layer model and support XFDU-ish [extensible data packaging format] unpacking specification" [1].

References:

1. METS - Metadata Encoding & Transmission Standard:
<<http://www.loc.gov/standards/mets/>>, METS Extenders:
<<http://www.loc.gov/standards/mets/mets-extenders.html>>
2. MODS - Metadata Object Description Schema:
<<http://www.loc.gov/standards/mods/>>
3. OAIS - Reference Model for an Open Archival Information System:
<<http://public.ccsds.org/publications/archive/650x0b1.pdf>>
4. W3C XML 1.0 (Second Edition): World Wide Web Consortium. *Extensible Markup Language (XML) 1.0, Second Edition*. <<http://www.w3.org/TR/2000/WD-xml-2e-20000814>>
5. W3C XML Schema: <<http://www.w3.org/XML/Schema>>
6. RFC 2045: N. Freed and N. Borenstein. *RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. 1996
<<http://www.ietf.org/rfc/rfc2045.txt>>
7. Repository Bridge: Automated Linkage of National and Institutional Repositories:
<<http://www.inf.aber.ac.uk/bridge/>>
8. "XML, SOAP and Binary Data" by A. Bosworth, D. Box, M. Gudgin, M. Nottingham, D. Orchard, J. Schlimmer: <<http://www.xml.com/lpt/a/2003/02/26/binaryxml.html>>
9. "Towards an Archival Information Package for Audiovisual Materials", Morgan Cundiff
<<http://www.rlg.org/longterm/forum02/cundiff.html>>, slides 9/10
10. "DSpace METS Document Profile for Submission Information Packages", Robert Wolfe and William Reilly, MIT Libraries
<<http://cwspace.mit.edu/docs/xsd/METS/SIP/profilev0p9p1/metsqipv0p9p1.pdf>>
11. "Extension Schemas for the Metadata Encoding and Transmission Standard" - LoC Digital Audio-Visual Preservation Prototyping Projects
<<http://www.loc.gov/rr/mopic/avprot/metsmenu2.html>>
12. PREMIS – Preservation Metadata Implementation Strategies:
<<http://www.loc.gov/standards/premis/>>
13. Project *kopal* – Universal Object Format: an archiving and exchange format for digital objects
<http://kopal.langzeitarchivierung.de/medien_presse/kopal_Universal_Object_Format.pdf>