1. TITLE	2. IDENTIFICATION NUMBER
ARCHIVAL INFORMATION PACKAGE (AIP) DESIGN STUDY	LC-DAVRS-07

3. DESCRIPTION/PURPOSE

The Archival Information Package (AIP) Design Study examines the technical issues surrounding the design of an AIP for the digital preservation of audio-visual items. The goal of the report is to define in broad terms the content and structure of an AIP for potential use by the Library of Congress (LC).

4. APPROVAL DATE (YYMMDD)	5. OFFICE OF PRIMARY RESPONSIBILITY	6a. COTR	6b. AUTHOR
010102	M/B/RS	Carl Fleischhauer	UTA

7. BACKGROUND

This report documents a study of the technical issues surrounding the design of an Archival Information Package (AIP). It was prepared under Contract Number 00CLCDV4920 (continuation of 99CLCCT1097), Option Year 1 - Lot 3 - Task No. 7 of the Digital Audio-Visual Repository System (DAVRS) Prototyping Project. The Prototyping Project is a three-year effort to study the feasibility of developing a digital repository system for the audio-visual collections at the Library of Congress (LC).

The objectives of this study are:

- To analyze the technical issues related to the design of an Archival Information Package;
- To develop a preliminary design of an AIP for LC digital AV collections; and
- To provide guidance for the development of a small sample of AIPs for feasibility testing.

What is an AIP?

An AIP is the digital equivalent of an archival item such as a book, a record album, or a motion picture. It consists of multiple data files that contain the digitized content of the archival item. In addition to the data files, the AIP contains metadata that describes the structure, content, and meaning of the data files. The data files and metadata are packaged (encapsulated) either logically or physically as an entity. AIPs are used to transmit and/or store archival objects within a digital repository system.

8. APPROVAL LIMITATION (YYMMDD)	9a. APPLICABLE FORMS	9b. APPROVAL AUTHORITY
Limited Approval from 010102 through 010331	N/A	Carl Fleischhauer

10. REFERENCE DOCUMENTS

1. Reference Model for an Open Archival Information System (OAIS); May 1999
2. The Making of America II Testbed Project White Paper Version 2.0, Jerome McDonough; September 1998
3. Collection-Based Long-Term Preservation, Reagan Moore; June 1999
4. Metadata Standards and the Making of America II DTD, Jerome McDonough; February 2001
5. Preservation Metadata for Digital Objects, OCLC/RLG White Paper; January 2001
6. Metadata for Digital Preservation, The CEDARS Project; March 2000
7. MPEG-21 Digital Item Declaration WD (v1.0), ISO/IEC JTC 1/SC 29/WG 11/N3825; January 2001
8. EBU P/Meta WP1 Metadata Scheme “Working Draft 01”, European Broadcast Union; March 2000
9. Technical Metadata for Digital Still Images, National Information Standards Organization; July 2000
10. A Comparison of Schemas for Video Metadata Representation, Hunter and Armstrong; January 2001

11. DISTRIBUTION STATEMENT

Library of Congress

Archival Information Package (AIP)

Design Study

LC-DAVRS-07

Final Draft

Acknowledgements

This study was conducted in collaboration with LC staff. The author wishes to acknowledge the contributions and assistance of the following individuals:

Martha Anderson, National Digital Library Program
Caroline Arms, National Digital Library Program and Information Technology Services
Ardie Bausenbach, Automation Planning and Liaison Office, Library Services
Dwight Beeson, Information Technology Services
Morgan Cundiff, Network Development and MARC Standards Office, Library Services
Gene DeAnna, Motion Picture, Broadcasting, and Recorded Sound Division, Library Services
Carl Fleischhauer, National Digital Library Program
Barbara Humphrys, Motion Picture, Broadcasting, and Recorded Sound Division, Library Services
Jane Mandelbaum, Information Technology Services
Ron Murray, Preservation Reformatting Division, Library Services
Mark Roosa, Preservation Directorate, Library Services
Nancy Seeger, Motion Picture, Broadcasting, and Recorded Sound Division, Library Services
Dick Thaxter, Motion Picture, Broadcasting, and Recorded Sound Division, Library Services
Barbara Tillett, Electronic Resources and Integrated Library System Program
Michael Winters, Information Technology Services

The author is particularly grateful to Carl Fleischhauer, Dick Thaxter, Morgan Cundiff, and Caroline Arms for their many hours of review and editing. Without their support this report could not have been prepared.

Table of Contents

1. Overview and Summary — Pages 5-10
   1.1. Introduction
   1.2. AIP Preliminary Design Summary
2. AIP Component-Level Design — Pages 11-29
   2.1. Metadata
   2.2. Data
   2.3. Packaging
3. Analysis and Recommendations — Pages 30-42
   3.1. Metadata
   3.2. Data
   3.3. Packaging
   3.4. Conclusions and Recommendations

Appendix A – METS Primary Schema — Pages 43-51
Appendix B – Survey of Related Projects — Pages 52-58
Appendix C – Related Links — Pages 59-60
Appendix D – Glossary — Pages 61-70

Section 1 – Overview and Summary

1.1 Introduction

The Library of Congress (LC) is investigating alternatives for preserving selected portions of its audio-visual collections as digital archival objects. This report examines the possibility that these digital archival objects could be instantiated as Archival Information Packages (AIPs). Included in the report is a preliminary design of an AIP. The report was prepared under Library of Congress Contract Order Number 00CLCDV4920 (continuation of 99CLCCT1097).

The goal of the report is to define in broad terms the content and structure of an AIP for potential use by LC. The objectives are:

- analyze the technical issues related to the design of an AIP;
- develop a preliminary design of an AIP for digital audio-visual items; and
- provide guidance for the development of a small sample of AIPs for feasibility testing.

The report is a product of the Digital Audio-Visual Repository System (DAVRS) Prototyping Project. The Prototyping Project is a multi-year research effort to study the feasibility of developing a digital repository capability for audio-visual collections held by the LC. The Prototyping Project embraces the general structure of a digital repository presented in the Open Archival Information System (OAIS) reference model. The Prototyping Project focuses primarily on the OAIS Ingestion and Access subsystems since they will require special features or capabilities to accommodate audio-visual content. Of particular interest is the definition of interfaces between the OAIS subsystems. Since the Library may develop individual digital repository subsystems at different times, interface definition will be especially important for the institution's overall repository design.

Note: The terminology in the OAIS Reference Model and some of the other sources cited in Appendix B often differs, and the same word may carry different meanings. The OAIS Reference Model definitions tend to be similar to those generally found in the field of computer science whereas other projects may share usage with archival and library science. For example, the term collection *has a very specific meaning for archivists that relates to the provenance of, say, a group of manuscripts or photographs. Librarians may use the term* collection *for the entire corpus of books held by an institution or to name, say, a* class *of books that is unified by some intellectual element. This latter use is similar to that of computer scientists, although the computer-relevant characteristics that bind a class differ from those relevant to the librarian. This document generally adheres to OAIS usage, and thus with the field of computer science. In some cases, however, as is the case with the word* collection*, library- or archives-oriented usage has been followed. A glossary is provided as Appendix D.*

The AIP not only plays a critical role in the interfaces between the modules of a digital repository but also represents a holding of the Library of Congress. The content of an AIP would be an asset of the Library that embodies value to its custodians and to society equivalent to the value of a physical book, record album, or motion picture. Just as physical collections must be housed safe from flood, theft, fire, and deterioration due to an adverse environment, Archival Information Packages must be stored and managed in a digital repository in a reliable manner for the long term. Responsibility for preserving this digital content is shared by two administrative units: the curators who are the keepers of the collections and the service providers who operate the systems in which they are preserved.

In parallel with the approach presented by this report, the Prototyping Project is also processing and storing digital content within the Library's computing environment using current storage management techniques. This activity will continue at the same time that the proposals in this document are under discussion and being tested.

An AIP contains both data files (essence bitstreams) and metadata. It is packaged as a single entity for transmission or archival storage. The Prototyping Project planners view the AIP primarily as a transmission and long-term storage format. For access and presentation, an AIP may be de-constructed or otherwise treated to meet the needs of data management and the user community.

The metadata component of the AIP will be embodied in a set of XML schemas being developed as a successor to the Making of America II (MOA2) DTD. This new set of schemas was recently named the Metadata Encoding and Transfer Standard (METS).

The METS developers (including LC participants) see a valuable role for this schema to support the current needs of libraries in object creation and management. This report suggests extending METS to include additional metadata to serve long-term digital preservation.

The METS will be structured as a Primary XML schema, featuring a File Group, a Structural Map, and “placeholders or sockets” for the insertion of Extension Schemas that provide descriptive and administrative information, e.g., schema versions of MARC or Dublin Core, NISO imaging metadata, AES audio metadata, and the like. Since these Extension Schemas are emergent, this report presents early versions of several local Library of Congress schemas proposed for use in interim AIP experiments.

1.2 AIP Preliminary Design Summary

The proposed AIP preliminary design is based upon the architectural concepts of the OAIS Reference Model and the metadata model specified in the METS. The following sections view the AIP design from four views: physical, logical, structural, and system. Together, they provide an overall understanding of the design of an AIP.

1.2.1 Physical View

From a physical perspective the AIP consists of three components: metadata, data, and packaging. Each component consists of one or more files. The metadata component consists of XML schemas containing information describing the archival object. The data component consists of all the data files (essence bitstream s) that comprise the archival object. The packaging component encapsulates the metadata and data components, creating a single entity or “self-extracting archive” that is the AIP. Exhibit 1-1 illustrates a physical view of the AIP architecture in its simplest form.

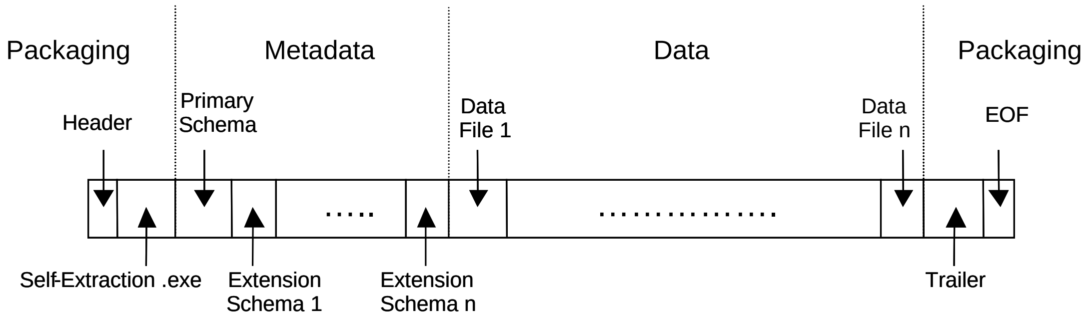


Exhibit 1-1. AIP Physical View.

1.2.2 Logical View

From a logical perspective the AIP is organized and defined by its metadata documented in XML form. The XML conforms (is validated) by one or more XML schemas. Additional sources of metadata can be referenced from within the XML. The logical view illustrated in Exhibit 1-2 provides an overview of the proposed XML schemas and reference structure of the metadata component of an AIP.

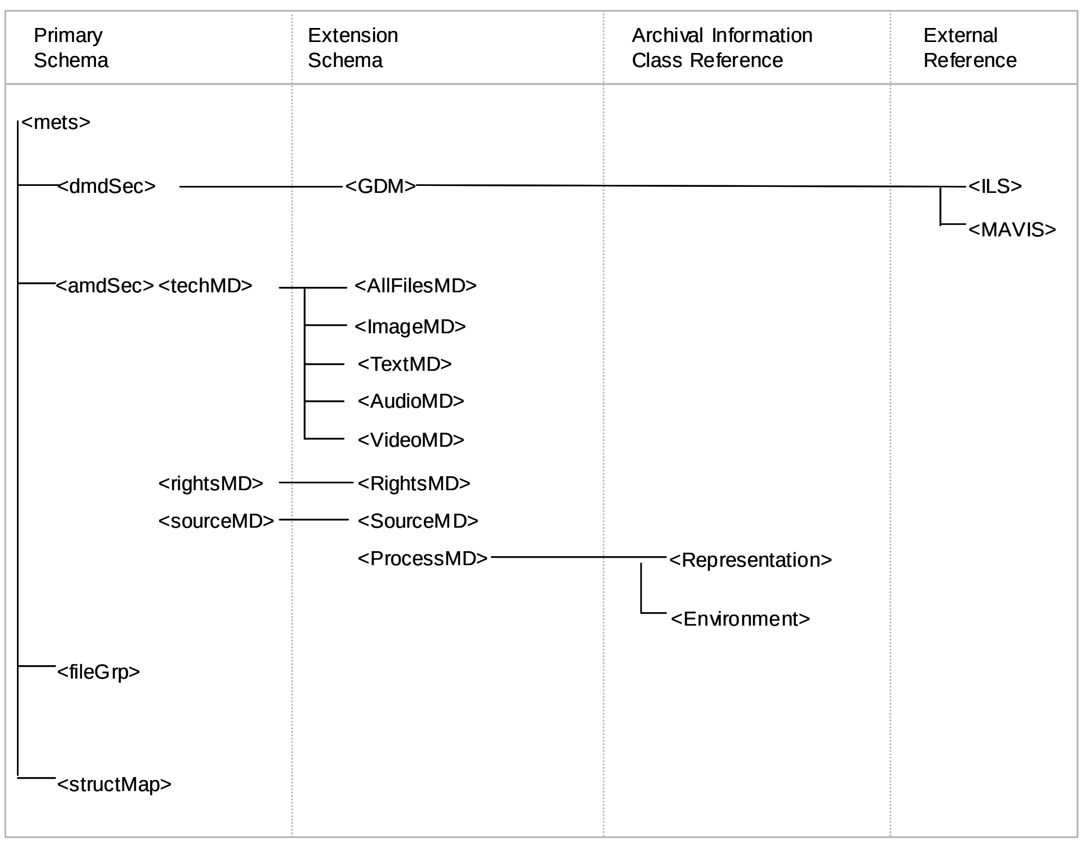


Exhibit 1-2. AIP Logical View.

1.2.2.1 Primary Schema

The Primary Schema contains only basic metadata about the archival object and is the most recent version of the emerging Digital Library Federation METS standard. The fundamental purpose of the Primary Schema is to define the structure and categories of metadata required to exchange AIPs with external organizations. Most of the detailed metadata is contained within the Extension Schemas. Appendix A contains a listing of the METS Primary Schema.

1.2.2.2 Extension Schemas

The Extension Schemas are separate namespace partitions referenced from within the Primary Schema. Extension Schemas contain metadata that is either too detailed, volatile, or parochial to be contained in the Primary Schema. There are nine (9) Extension Schemas currently defined in the AIP metadata model proposed herein: GDM, AllFilesMD, ImageMD, TextMD, AudioMD, VideoMD, RightsMD, SourceMD, and ProcessMD. Listings of the proposed Extension Schemas can be found on the Prototyping Project web site at http://lcweb.loc.gov/rr/mopic/avprot/avlcdocs.html#md.

1.2.2.3 Archival Information Class (AIC) References

An Archival Information Class (AIC) is a type of AIP that contains metadata pertaining to a class of AIPs. All AIPs that are members of a particular class inherit the metadata of that AIC. The structure of an AIC is identical to that of an AIP. The data files that comprise an AIC contain metadata for a class of AIPs. The use of AICs offers the following advantages:

- the type and amount of information contained in an AIC is discretionary;
- large volumes of metadata can be accommodated easily;
- the information contained in an AIC is preserved in the same archive as its AIPs;
- certain types of metadata for an entire class of AIPs need only be stored once;
- as a new class is created, a new AIC can be added without significant changes to its AIPs.

The proposed AIP metadata model defines two types of AICs: Representation, and Environment.

1.2.2.4 External References

Systems or files external to a digital repository (such as ILS or MAVIS at LC) can be referenced from within the Primary Schema or the Extension Schemas. External references are useful as links to additional information sources. The most common external reference is to catalog systems or finding aids used in the discovery process.

1.2.3 Structural View

The structural view describes the navigational perspective of an AIP. It provides one or more maps by which the user can navigate within an archival object. It associates the physical nature of the original source item, from which the archival object was created, with the metadata and data that comprise the AIP. Exhibit 1-3 illustrates an example of a user view of the AIP. The object pictured represents a digital reproduction of a phonodisc album, with digital audio bitstreams representing the sound elements in the original and digital images representing the original printed elements.

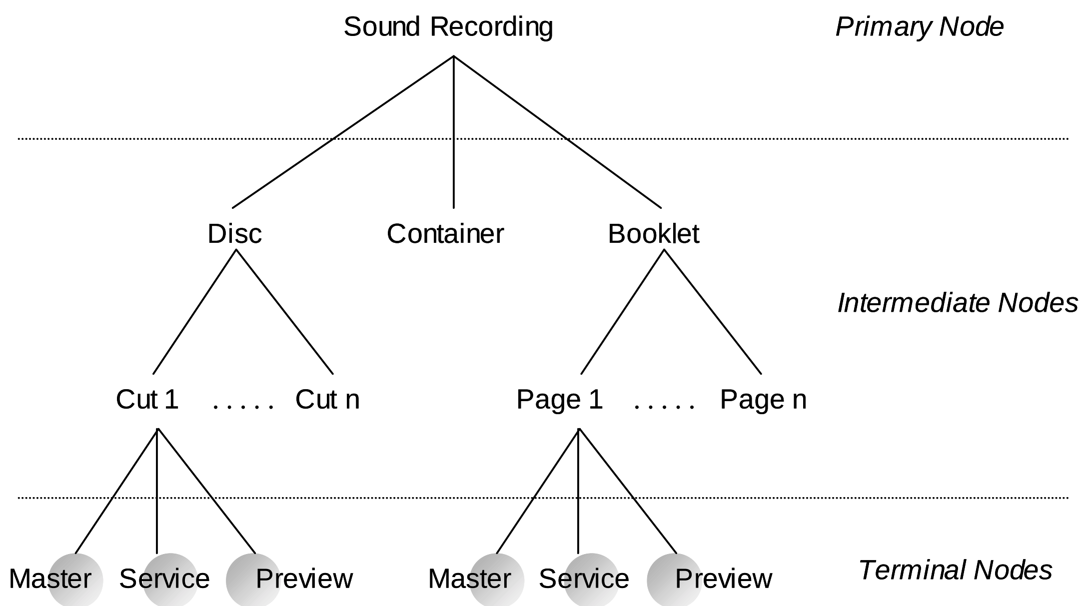


Exhibit 1-3. AIP Structural View.

An AIP structural view is most often hierarchical. At the top of the hierarchy is the Primary Node that corresponds to the archival object itself. The Primary Node decomposes into one or more levels of Intermediate Nodes that represent sub-objects of the archival object. The lowest level of decomposition is the Terminal Node, from which there is no further decomposition. The level of decomposition of Intermediate Nodes is unlimited. Frequently, Terminal Nodes will correspond to individual files. However, Terminal Nodes may also be defined at the sub-file level as well. Using this generic hierarchical structure any number of user views can be created. The <structMap> element of the METS Primary Schema defines the structural view of an archival object.

1.2.4 System View

The AIP was designed in the context of an open archival information system. However, it should be noted that an intrinsic value of the AIP concept is infrastructure independence. If an AIP is properly formed, it should be able to reside and be rematerialized in a variety of computing environments. Exhibit 1-4 illustrates an example of a system view of an AIP.

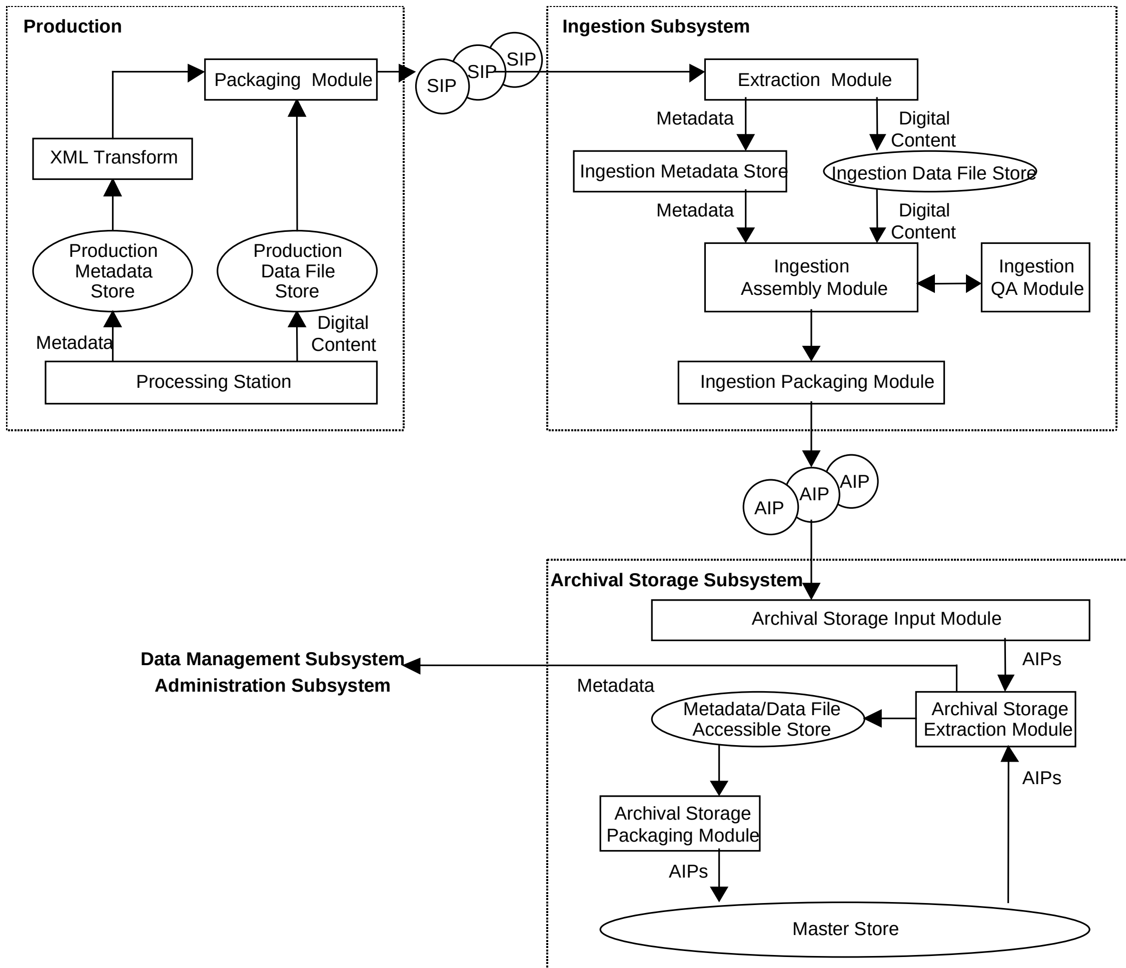


Exhibit 1-4. AIP System View.

Section 2 - AIP Component-Level Design

Section 2 describes the preliminary component-level design of an AIP. It is intended for use as the starting point for the development of a small set of sample AIPs, derived from archival items chosen for use in the Prototyping Project. Section 2.1 describes the metadata component design. Section 2.2 describes the data component design. Section 2.3 describes the packaging component design.

2.1 Metadata Component

As described in Section 1, the metadata component of the AIP is organized as a set of XML schemas consisting of a Primary Schema and multiple Extension Schemas. The METS metadata schemas are currently under construction. The following section proposes a metadata model based on these emerging METS schemas. The metadata is presented graphically in XMLSpy 3.5 design view format. Text views of the Primary and Extension Schemas can be found in Appendix A and at http://lcweb.loc.gov/rr/mopic/avprot/avlcdocs.html#md along with additional information regarding metadata used by the Prototyping Project.

It should be noted that the specification of metadata standards at the LC is a work in progress based upon changing requirements within the LC and rapidly evolving standards for digital preservation metadata throughout the library community. An attempt was made in this report to incorporate current thinking; however, it is inevitable and desirable that changes will occur in the future.

2.1.1 Primary Schema

The Primary Schema is used to identify the archival object, to define the structure of the archival object, to inventory the files that comprise the archival object, and to reference Extension Schemas that further describe the archival object. The version of the Primary Schema presented in this report was developed by Jerome McDonough of New York University and released to the METS listserve in May 2001. The METS Primary Schema has four major elements: <dmdSec>, <amdSec>, <fileGrp>, and <structMap>. Although final agreement about the type of metadata to be contained in the Primary Schema has not yet been reached, in general <dmdSec> contains descriptive (intellectual) metadata of the type used in discovery, <amdSec> contains administrative metadata needed to manage and preserve the archival object, <fileGrp> contains information about the inventory of files that comprise the archival object, and <structMap> defines the structure of the archival object, primarily from a presentation perspective.

The <dmdSec> and <amdSec> elements are essentially pointers to Extension Schemas. The <fileGrp> and <structMap> elements are more sophisticated versions of templates originally developed at the University of California Berkeley for the Making of America II DTD. Exhibit 2-1 contains a high-level graphic depiction of the METS Primary Schema.

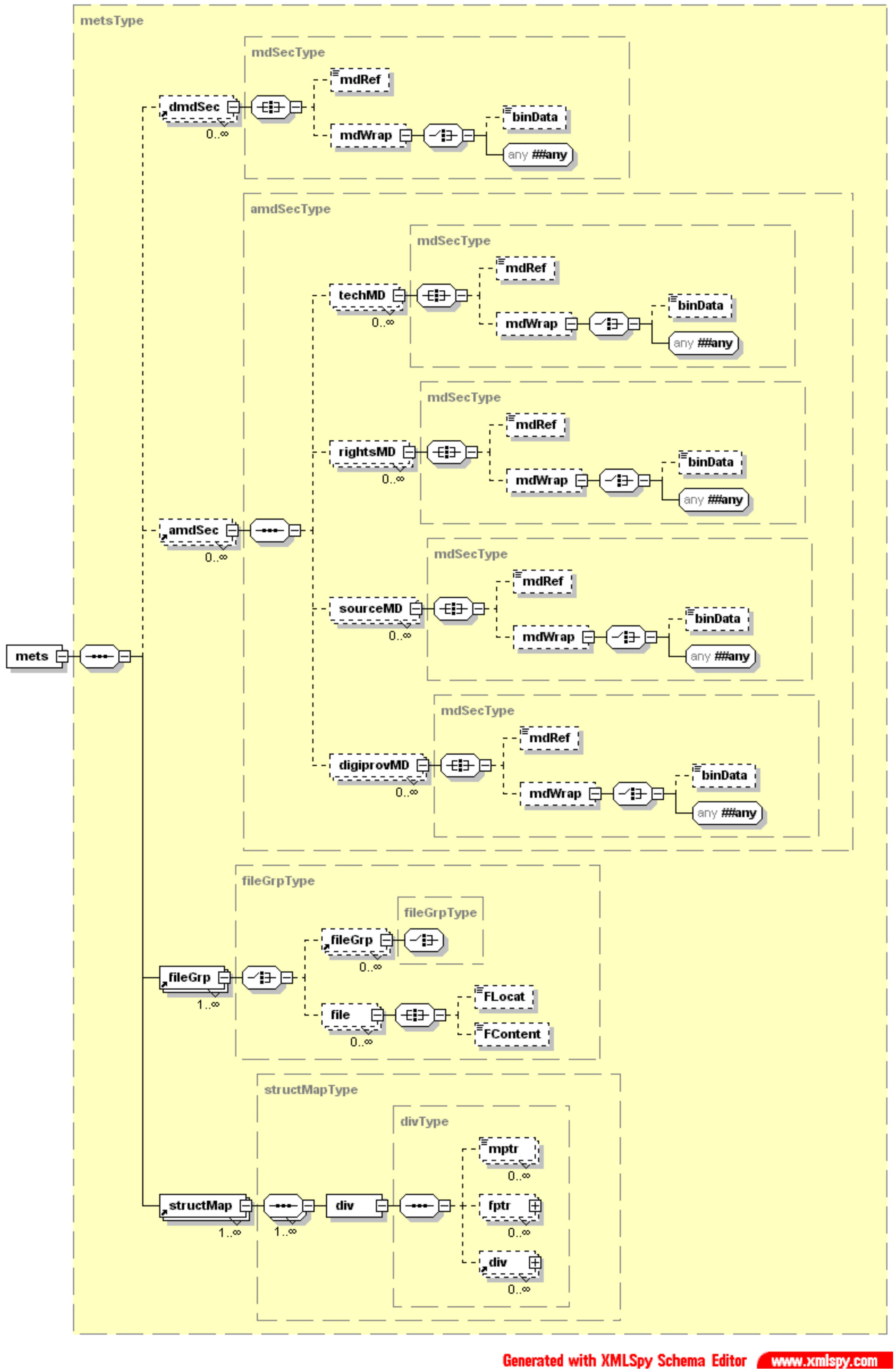


Exhibit 2-1. METS Primary Schema.

The <dmdSec> element references a sub-element called <mdSec> that defines a “placeholder or socket” allowing the Primary Schema to be linked to Extension Schemas or external data sources. This construct provides a powerful tool for the customization of descriptive metadata by individual organizations.

The <amdSec> element defines the administrative metadata for an AIP. The “socket” construct is also applied to each of the sub-elements of <amdSec> (i.e. <techMD>, <rightsMD>, and <sourceMD>). The Extension Schemas for technical, rights, and source metadata will be described later in Section 2.1.2.

The <fileGrp> element is used to define the inventory of files contained in an AIP. Its recursiveness allows files to be grouped for purposes of administration. The <file> element contains sub-elements and attributes that define the location and optionally the content of each file in the AIP.

The <structMap> element is used to define the structural view of an AIP. The METS standard represents the archival object structurally as a series of nested <div> elements, that is, as a hierarchy (e.g., a sound recording, which is composed of discs, container, and booklet, which are composed of cuts or pages, which are represented by audio files or image files, etc.). Every <div> node in the structural map hierarchy may be connected (via <mptr> or <fptr> sub-elements) to content file(s) that represent that div's portion of the whole document.

2.1.2 Extension Schemas

At the time this document went to press, the Prototyping Project had identified or developed nine (9) Extension Schemas for use in the AIP metadata model as listed below. The GDM Extension Schema was developed by Rick Beaubien at the University of California, Berkeley. The remaining eight (8) Extension Schemas were developed as a task of the Prototyping Project. Listings of the proposed Extension Schemas can be found on the Prototyping Project web site at http://lcweb.loc.gov/rr/mopic/avprot/avlcdocs.html#md.

- GDM
- AllFilesMD
- ImageMD
- TextMD
- AudioMD
- VideoMD
- RightsMD
- SourceMD
- ProcessMD

2.1.2.1 GDM Extension Schema

The GDM Extension Schema contains descriptive metadata about the archival object. The GDM Extension Schema presented below was developed by Rick Beaubien at the University of California, Berkeley. Exhibit 2-2 contains a high-level graphic depiction of the GDM Extension Schema.

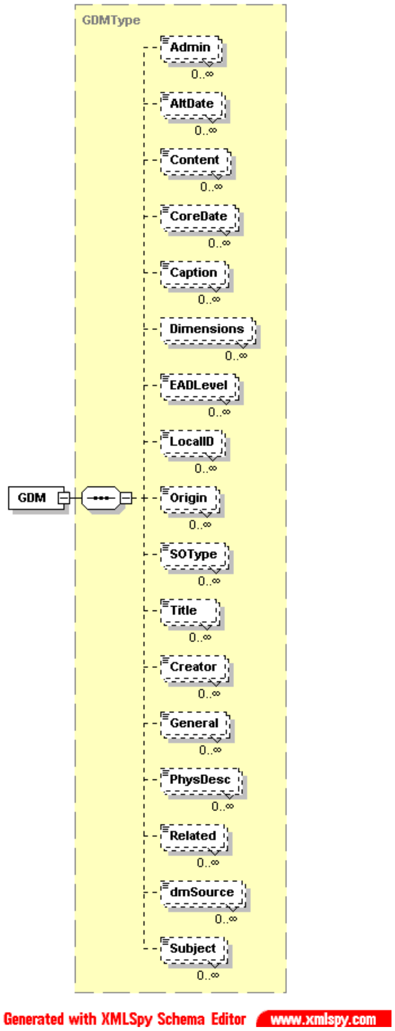
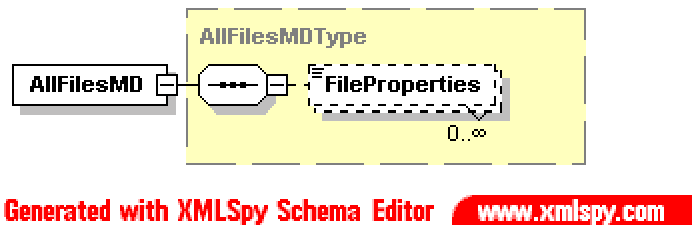


Exhibit 2-2. GDM Extension Schema.

The elements listed in the GDM Extension Schema were derived from the Making of America II Project DTD. The sub-elements and attributes subsidiary to the elements depicted above provide a flexible and varied template for the intellectual description of archival objects at the Primary, Intermediate, and Terminal Node levels. It should be noted however that the GDM Extension Schema is but one of many choices for descriptive metadata templates. Other forms such as Dublin Core could be substituted or one or more customized schemas could be developed.

2.1.2.2 AllFilesMD Extension Schema

The AllFilesMD Extension Schema defines file properties that are applicable to all digital files and is used to manage and access the files that comprise the AIP. Exhibit 2-3 contains a high-level graphic depiction of the AllFilesMD Extension Schema and a list of the attributes that comprise the file properties.



FileProperties

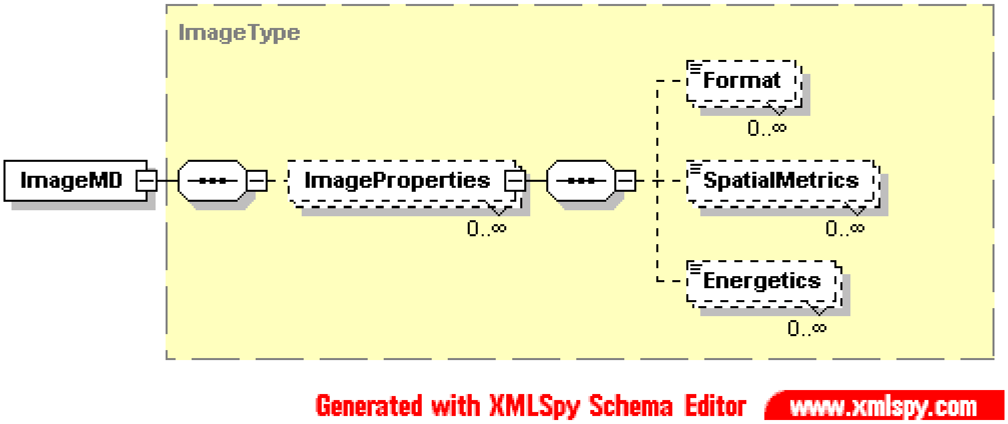
```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="SIZE" type="xsd:long" use="optional"/>
<xsd:attribute name="CREATE_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="CREATE_TIME" type="xsd:time" use="optional"/>
<xsd:attribute name="MODIFY_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="MODIFY_TIME" type="xsd:time" use="optional"/>
<xsd:attribute name="ACCESS_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="ACCESS_TIME" type="xsd:time" use="optional"/>
<xsd:attribute name="CHECKSUM_VALUE" type="xsd:long" use="optional"/>
<xsd:attribute name="CHECKSUM_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="CHECKSUM_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="ENCRYPTION" type="xsd:string" use="optional"/>
<xsd:attribute name="WATERMARK" type="xsd:string" use="optional"/>
<xsd:attribute name="COMPRESSION" type="xsd:string" use="optional"/>
<xsd:attribute name="USE" type="xsd:string" use="optional"/>
<xsd:attribute name="GENERATION" type="xsd:string" use="optional"/>
<xsd:attribute name="CURRENT" type="xsd:boolean" use="optional"/>

<xsd:attribute name="FILE_IDREF" type="xsd:IDREF" use="optional"/>
<xsd:attribute name="MEDIA_IDREF" type="xsd:IDREF" use="optional"/>
<xsd:attribute name="SOURCE_IDREF" type="xsd:IDREF" use="optional"/>
<xsd:attribute name="PRESERVATION_IDREF" type="xsd:IDREF" use="optional"/>
<xsd:attribute name="AUXILIARY_ADMIN_IDREFS" type="xsd:IDREFS" use="optional"/>
```

Exhibit 2-3 AllFilesMD Extension Schema.

Care was taken to avoid the inclusion of media-specific file properties that are addressed in separate media-specific Extension Schemas. In addition to the file properties attributes, there are a set of IDREF and IDREFS that reference tags (xsd:ID) within the Primary and other Extension Schemas that contain metadata related to the file. These references facilitate chaining between various metadata elements of the Primary and Extension Schemas during presentation.

2.1.2.3 ImageMD Extension Schema

The ImageMD Extension Schema contains administrative metadata that provides detailed technical information about an image file or a group of image files. The specific elements and attributes were derived from the emerging NISO image metadata standard. The metadata in the Image Extension Schema provides the detailed technical information required to support reformatting and rematerialization of the image file(s), and may also support future migration or system emulation processes. Exhibit 2-4 contains a high-level graphic depiction of the ImageMD Extension Schema and a list of the attributes.



Format

```
<xsd:attribute name="SEGMENT_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="STRIP_OFFSETS" type="xsd:integer" use="optional"/>
<xsd:attribute name="ROWS_PER_STRIP" type="xsd:integer" use="optional"/>
<xsd:attribute name="STRIP_BYTE_COUNTS" type="xsd:integer" use="optional"/>
<xsd:attribute name="TILE_WIDTH" type="xsd:integer" use="optional"/>
<xsd:attribute name="TILE_LENGTH" type="xsd:integer" use="optional"/>
<xsd:attribute name="TILE_OFFSETS" type="xsd:integer" use="optional"/>
<xsd:attribute name="TILE_BYTE_COUNTS" type="xsd:integer" use="optional"/>
<xsd:attribute name="PLANAR_CONFIGURATION" type="xsd:string" use="optional"/>
<xsd:attribute name="ORIENTATION" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPLAY_ORIENTATION" type="xsd:string" use="optional"/>
<xsd:attribute name="PREFERRED_PRESENTATION" type="xsd:IDREF" use="optional"/>
```

SpatialMetrics

```
<xsd:attribute name="IMAGE_WIDTH" type="xsd:string" use="optional"/>
<xsd:attribute name="IMAGE_LENGTH" type="xsd:string" use="optional"/>
<xsd:attribute name="X_SAMPLING_FREQUENCY" type="xsd:string" use="optional"/>
<xsd:attribute name="Y_SAMPLING_FREQUENCY" type="xsd:string" use="optional"/>
<xsd:attribute name="SAMPLING_FREQUENCY_UNIT" type="xsd:string" use="optional"/>
<xsd:attribute name="SAMPLING_FREQUENCY_PLANE" type="xsd:string" use="optional"/>
```

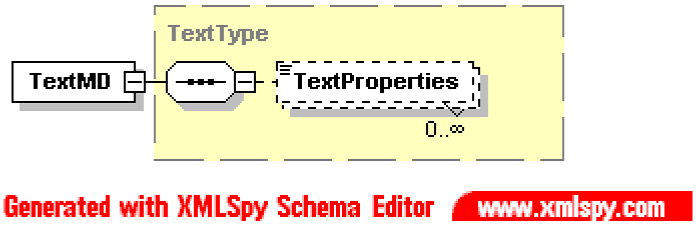
Energetics

```
<xsd:attribute name="BITS_PER_SAMPLE" type="xsd:integer" use="optional"/>
<xsd:attribute name="SAMPLES_PER_PIXEL" type="xsd:integer" use="optional"/>
<xsd:attribute name="EXTRASAMPLES" type="xsd:string" use="optional"/>
<xsd:attribute name="COLOR_MAP" type="xsd:string" use="optional"/>
<xsd:attribute name="GRAY_RESPONSE_CURVE" type="xsd:string" use="optional"/>
<xsd:attribute name="GRAY_RESPONSE_UNIT" type="xsd:string" use="optional"/>
<xsd:attribute name="WHITE_POINT" type="xsd:string" use="optional"/>
<xsd:attribute name="PRIMARY_CHROMATICITIES" type="xsd:string" use="optional"/>
```

Exhibit 2-4. ImageMD Extension Schema.

2.1.2.4 Text Extension Schema

The TextMD Extension Schema contains metadata that describes a text file or group of text files. The elements listed here were derived from fields contained the Prototyping Project metadata database. In general the LC uses markup language for digital text file storage and preservation. As XML is emerging as a nationwide standard for document formatting, it is likely that text in the AIP will be stored as XML or SGML files. Exhibit 2-5 contains a high-level depiction of the TextMD Extension Schema and a list of the attributes.



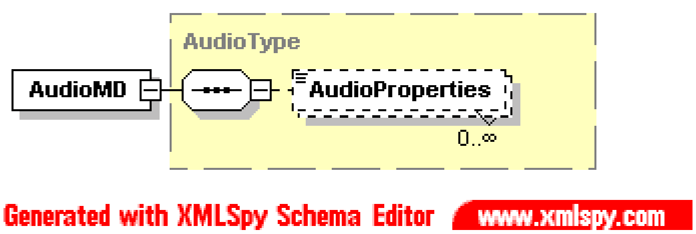
TextProperties

```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="ENCODING" type="xsd:string" use="optional"/>
<xsd:attribute name="MARKUP" type="xsd:string" use="optional"/>
<xsd:attribute name="WORD_PROCESSOR" type="xsd:string" use="optional"/>
<xsd:attribute name="NOTE" type="xsd:string" use="optional"/>
<xsd:attribute name="ADMIN_IDREF" type="xsd:IDREF" use="optional"/>
```

Exhibit 2-5. TextMD Extension Schema.

2.1.2.5 Audio Extension Schema

The AudioMD Extension Schema contains administrative metadata that provides detailed technical information about an audio file or a group of audio files. The elements and attributes below were derived from fields in the Prototyping Project metadata database. The data in the AudioMD Extension Schema is intended to support reformatting and rematerialization of the audio file(s) contained in the archival object, and may also support migration or system emulation processes. Exhibit 2-6 contains a graphic depiction of the AudioMD Extension Schema and a list of attributes.



AudioProperties

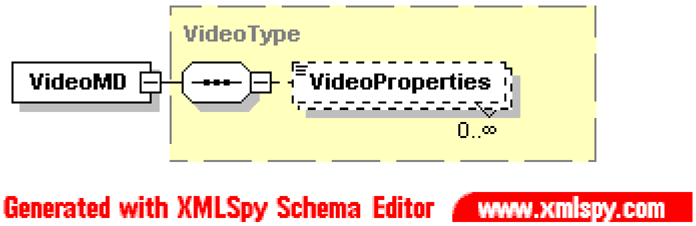
```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="BITS_PER_SAMPLE" type="xsd:string" use="optional"/>
<xsd:attribute name="SAMPLING_FREQUENCY" type="xsd:string" use="optional"/>
<xsd:attribute name="DATA_RATE_FIXED" type="xsd:string" use="optional"/>
<xsd:attribute name="DATA_RATE_VARIABLE" type="xsd:string" use="optional"/>
<xsd:attribute name="DURATION" type="xsd:string" use="optional"/>
```

```
<xsd:attribute name="EQUALIZATION" type="xsd:string" use="optional"/>
<xsd:attribute name="ADMIN_IDREF" type="xsd:IDREF" use="optional"/>
```

Exhibit 2-6. AudioMD Extension Schema.

2.1.2.6 Video Extension Schema

The VideoMD Extension Schema contains metadata that describes a video file or group of video files. The elements and attributes below were derived from fields in the Prototyping Project metadata database. The project team notes that these attributes are provisional, and that the group plans to consult with video engineering specialists to improve the identification and definitions for the proposed metadata. The data in the VideoMD Extension Schema is intended to support reformatting and rematerialization of the video file(s) contained in the archival object, and may also support migration or system emulation processes. Exhibit 2-7 contains a graphic depiction of the VideoMD Extension Schema and a list of attributes.



<u>VideoProperties</u>

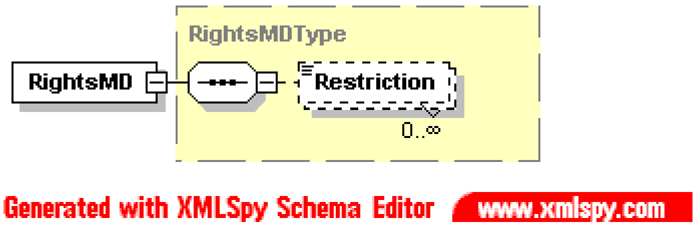
```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="VIDEO_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="COMPRESSION_CODEC" type="xsd:string" use="optional"/>
<xsd:attribute name="DATA_RATE" type="xsd:decimal" use="optional"/>
<xsd:attribute name="DURATION" type="xsd:time" use="optional"/>
<xsd:attribute name="FRAME_RATE" type="xsd:integer" use="optional"/>
<xsd:attribute name="PIXELS_HORIZONTAL" type="xsd:integer" use="optional"/>
<xsd:attribute name="PIXELS_VERTICAL" type="xsd:integer" use="optional"/>
<xsd:attribute name="SOUND_CHANNEL_CONFIGURATION" type="xsd:string" use="optional"/>
<xsd:attribute name="SOUND_CHANNEL_INFO" type="xsd:string" use="optional"/>
<xsd:attribute name="SOUND_FORMAT" type="xsd:string" use="optional"/>
<xsd:attribute name="ADMIN_IDREF" type="xsd:IDREF" use="optional"/>
```

Exhibit 2-7. VideoMD Extension Schema.

The ImageMD, TextMD, AudioMD, and VideoMD Extension Schemas each include an ADMIN_IDREF attribute that allows a backward reference to an AllFilesMD xsd:ID. Via that backward reference they can be directly linked to other AllFilesMD administrative metadata and thereby indirectly linked to related descriptive and file inventory metadata.

2.1.2.7 RightsMD Extension Schema

The RightsMD Extension Schema contains metadata that describes access and copyright information about the archival object. For materials considered to be protected by copyright, the Library produces preservation copies but limits access to these copies, in some cases to a single reading room. Thus the current metadata requirement has to do with providing support for *access* *management* rather than *rights management*. Future activities at the Library, however, may have a broader scope and additional metadata may be required. This schema also includes places to record ownership and copyright information, to serve future needs or to serve the needs of other applications. Exhibit 2-8 provides examples of the metadata contained in the RightsMD Extension Schema.



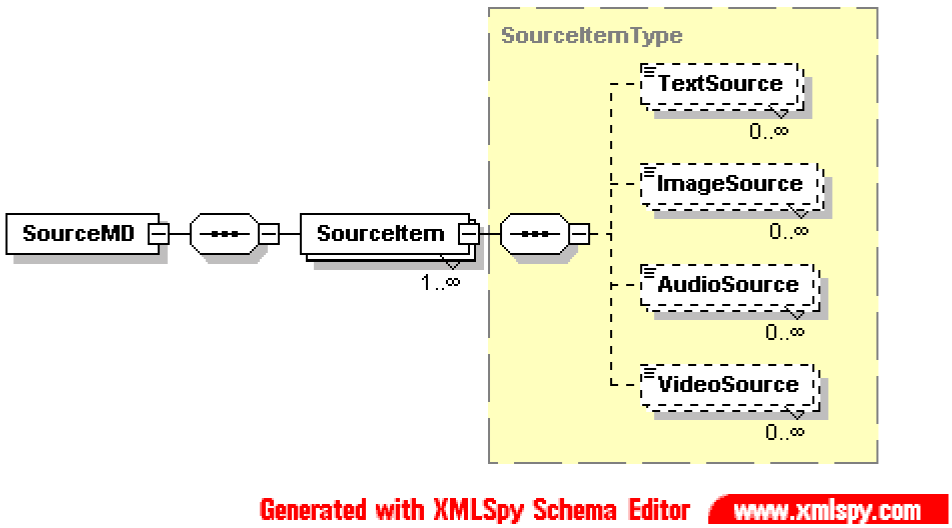
Restrictions

```
<xsd:enumeration value="unrestricted_copyright_expired"/>
<xsd:enumeration value="unrestricted_deed_of_gift"/>
<xsd:enumeration value="unrestricted_US_government"/>
<xsd:enumeration value="unrestricted_LC_review_unrestricted"/>
<xsd:enumeration value="restricted_LC_review_low risk"/>
<xsd:enumeration value="restricted_LC_review_high risk"/>
<xsd:enumeration value="restricted_copyrighted_permission_not_sought"/>
<xsd:enumeration value="restricted_copyrighted_permission_granted"/>
<xsd:enumeration value="restricted_copyrighted_permission_denied"/>
<xsd:enumeration value="restricted_donor_deed_of_gift"/>
<xsd:enumeration value="restricted_classified"/>
<xsd:enumeration value="restricted_non_copyright_issue"/>
<xsd:enumeration value="unrestricted_copyright_expired"/>
<xsd:enumeration value="restricted_recsound_permission_not_sought"/>
<xsd:enumeration value="restricted_folklife_permission_not_sought"/>
<xsd:enumeration value="restricted_folklife_permission_denied"/>
<xsd:enumeration value="restricted_recsound_permission_granted"/>
<xsd:enumeration value="restricted_co_in_process"/>
<xsd:enumeration value="restricted_co_not_released"/>
<xsd:enumeration value="restricted_co_released_no_access"/>
<xsd:enumeration value="restricted_co_released_selectable"/>


<xsd:attribute name="REGISTRATION_NUMBER" type="xsd:string" use="optional"/>
<xsd:attribute name="HOLDER" type="xsd:string" use="optional"/>
<xsd:attribute name="CREDIT_LINE" type="xsd:string" use="optional"/>
<xsd:attribute name="COPY_RESTRICTION" type="xsd:string" use="optional"/>
<xsd:attribute name="DISTRIBUTION_RESTRICTION" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPLAY_RESTRICTION" type="xsd:string" use="optional"/>
<xsd:attribute name="RESTRICTION_EXPIRATION_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="RESTRICTION_TERRITORY" type="xsd:string" use="optional"/>
<xsd:attribute name="LICENSING" type="xsd:string" use="optional"/>
<xsd:attribute name="EXPLANATION" type="xsd:string" use="optional"/>
```

Exhibit 2-8. RightsMD Extension Schema.

2.1.2.8 SourceMD Extension Schema

The SourceMD Extension Schema contains detailed metadata about the source item(s) from which the archival object was created. Given that many source items, such as motion pictures and recorded sound, are composed of multiple sub-items that are captured and digitized as files of different media types, the source item is subdivided into <TextSource>, <ImageSource>, <AudioSource>, and <VideoSource> elements. Exhibit 2-9 contains a graphic depiction of the SourceMD Extension Schema and a list of attributes.



<u>TextSource</u>

```
<xsd:attribute name="ID" type="xsd:ID" use="required"/>
<xsd:attribute name="TRANSCRIBER" type="xsd:string" use="optional"/>
<xsd:attribute name="OCR" type="xsd:string" use="optional"/>
<xsd:attribute name="CONDITION" type="xsd:string" use="optional"/>
<xsd:attribute name="DIMENSIONS" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPOSITION" type="xsd:string" use="optional"/>
<xsd:attribute name="NOTE" type="xsd:string" use="optional"/>
```

<u>ImageSource</u>

```
<xsd:attribute name="ID" type="xsd:ID" use="required"/>
<xsd:attribute name="CONDITION" type="xsd:string" use="optional"/>
<xsd:attribute name="DIMENSIONS" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPOSITION" type="xsd:string" use="optional"/>
<xsd:attribute name="NOTE" type="xsd:string" use="optional"/>
```

<u>AudioSource</u>

```
<xsd:attribute name="ID" type="xsd:ID" use="required"/>
<xsd:attribute name="CONDITION" type="xsd:string" use="optional"/>
<xsd:attribute name="DIMENSIONS" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPOSITION" type="xsd:string" use="optional"/>
<xsd:attribute name="NOTE" type="xsd:string" use="optional"/>
<xsd:attribute name="BASE_MATERIAL" type="xsd:string" use="optional"/>
<xsd:attribute name="BINDER" type="xsd:string" use="optional"/>
<xsd:attribute name="BITS_PER_SAMPLE" type="xsd:string" use="optional"/>
<xsd:attribute name="CALIBRATION_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="CALIBRATION_INFO" type="xsd:string" use="optional"/>
<xsd:attribute name="CALIBRATION_LOCATION" type="xsd:string" use="optional"/>
<xsd:attribute name="CHANNEL_TRACK_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="CHANNEL_TRACK_MAPPING" type="xsd:string" use="optional"/>
<xsd:attribute name="DATA_RATE_FIXED" type="xsd:string" use="optional"/>
```

```
<xsd:attribute name="DATA_RATE_VARIABLE" type="xsd:string" use="optional"/>
<xsd:attribute name="DISK_SURFACE" type="xsd:string" use="optional"/>
<xsd:attribute name="DIGITAL" type="xsd:boolean" use="optional"/>
<xsd:attribute name="EQUALIZATION" type="xsd:string" use="optional"/>
<xsd:attribute name="FORMAT_SPECIFICATION_NAME" type="xsd:string" use="optional"/>
<xsd:attribute name="FORMAT_SPECIFICATION_INFO" type="xsd:string" use="optional"/>
<xsd:attribute name="DURATION" type="xsd:string" use="optional"/>
<xsd:attribute name="GROOVE" type="xsd:string" use="optional"/>
<xsd:attribute name="GUAGE" type="xsd:string" use="optional"/>
<xsd:attribute name="LENGTH" type="xsd:string" use="optional"/>
<xsd:attribute name="NOISE_REDUCTION" type="xsd:string" use="optional"/>
<xsd:attribute name="OXIDE" type="xsd:string" use="optional"/>
<xsd:attribute name="SAMPLING_FREQUENCY" type="xsd:string" use="optional"/>
<xsd:attribute name="SOUND_FIELD" type="xsd:string" use="optional"/>
<xsd:attribute name="SPEED" type="xsd:string" use="optional"/>
<xsd:attribute name="SPEED_ADJUSTMENT" type="xsd:string" use="optional"/>
<xsd:attribute name="STOCK_BRAND" type="xsd:string" use="optional"/>
<xsd:attribute name="TRACK_FORMAT" type="xsd:string" use="optional"/>
```

VideoSource

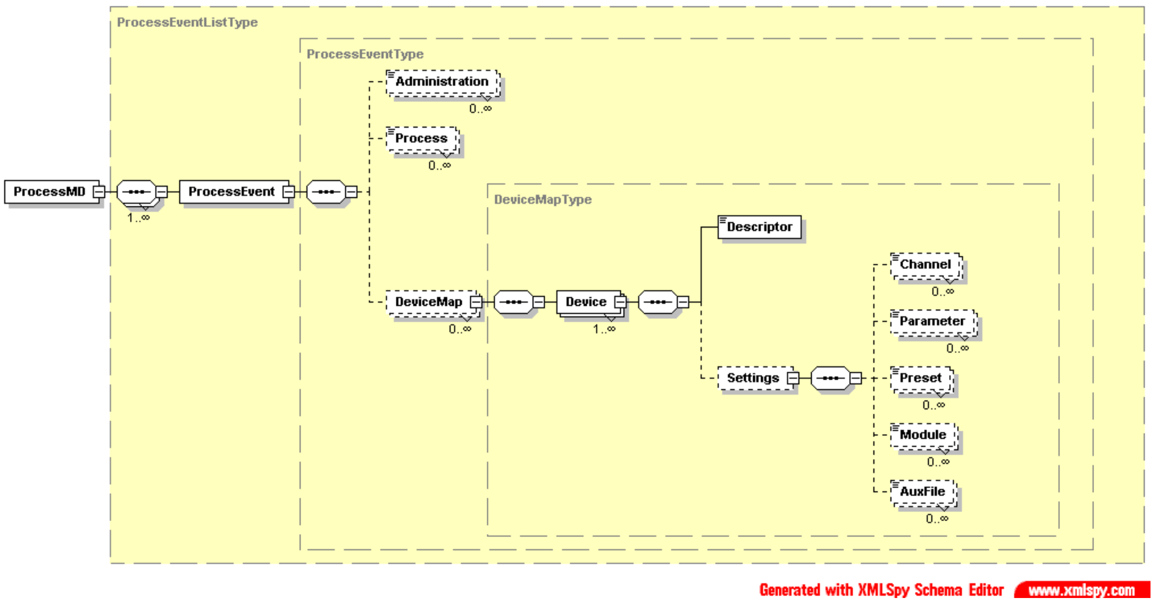
```
<xsd:attribute name="ID" type="xsd:ID" use="required"/>
<xsd:attribute name="CALIBRATION_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="CALIBRATION_INFO" type="xsd:string" use="optional"/>
<xsd:attribute name="CALIBRATION_LOCATION" type="xsd:string" use="optional"/>
<xsd:attribute name="CONDITION" type="xsd:string" use="optional"/>
<xsd:attribute name="DIMENSIONS" type="xsd:string" use="optional"/>
<xsd:attribute name="DISPOSITION" type="xsd:string" use="optional"/>
<xsd:attribute name="NOTE" type="xsd:string" use="optional"/>
```

Exhibit 2-9. SourceMD Extension Schema.

At the time this report was written the metadata for TextSource, ImageSource, and VideoSource were still under development and are therefore limited in scope.

2.1.2.9 ProcessMD Extension Schema

The ProcessMD Extension Schema contains information that documents the history of the archival object, including how it was made, who made it, and what changes have been made to it. This includes information about the treatment of that object or aspects of the production process not recorded elsewhere. In addition, this block of administrative metadata is also the location for documenting the history and transformations of the archival object through time. The extension schema represented in the following metadata list is a blend of (1) elements found in the Prototyping Project metadata database and (2) software release management methodology. Exhibit 2-10 contains a graphic depiction of the Provenance Extension Schema and a list of attributes.



ProcessEvent

```
<xsd:attribute name="ID" type="xsd:ID"/>
<xsd:attribute name="N" type="xsd:integer" use="optional"/>
<xsd:attribute name="NUMBER" type="xsd:string" use="optional"/>
<xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
<xsd:attribute name="ADMIN_IDREFS" type="xsd:IDREFS" use="optional"/>
<xsd:attribute name="MEDIA_IDREFS" type="xsd:IDREFS" use="optional"/>
<xsd:attribute name="SOURCE_IDREF" type="xsd:IDREF" use="optional"/>
```

Administration

```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="CAPTURE_PRODUCTION_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="ORGANIZATION" type="xsd:string" use="optional"/>
<xsd:attribute name="INDIVIDUAL" type="xsd:string" use="optional"/>
<xsd:attribute name="LOCATION" type="xsd:string" use="optional"/>
<xsd:attribute name="DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="TIME" type="xsd:time" use="optional"/>
```

Process

```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="TREATMENT_ACTION" type="xsd:string" use="optional"/>
<xsd:attribute name="TREATMENT_NOTE" type="xsd:string" use="optional"/>
<xsd:attribute name="PRESERVATION_INFO" type="xsd:string" use="optional"/>
<xsd:attribute name="REFORMAT_GUIDELINES" type="xsd:string" use="optional"/>
<xsd:attribute name="REFORMAT_METHOD" type="xsd:string" use="optional"/>
<xsd:attribute name="QA_INSPECTOR" type="xsd:string" use="optional"/>
<xsd:attribute name="QA_DATE" type="xsd:date" use="optional"/>
<xsd:attribute name="QA_PASSED" type="xsd:boolean" use="optional"/>
<xsd:attribute name="QA_NOTE" type="xsd:string" use="optional"/>
```

Descriptor

```
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="MODEL" type="xsd:string" use="optional"/>
<xsd:attribute name="SERIAL_NUMBER" type="xsd:string" use="optional"/>
```

```
<xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
<xsd:attribute name="DEVICE_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="INPUT_OUTPUT" type="xsd:string" use="optional"/>
```

Channel

```
<xsd:attribute name="CHANNEL_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="CHANNEL_TYPE" type="xsd:string" use="optional"/>
<xsd:attribute name="PATCHED_DEVICE_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="PATCHED_CHANNEL_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="CHANNEL_NOTE" type="xsd:string" use="optional"/>
```

Parameter

```
<xsd:attribute name="PARAMETER_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="PARAMETER_NAME" type="xsd:string" use="optional"/>
<xsd:attribute name="PARAMETER_TYPE" type="xsd:string" use="optional"/>
```

Preset

```
<xsd:attribute name="PRESET_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="PRESET_NAME" type="xsd:string" use="optional"/>
<xsd:attribute name="PRESET_PARAMETER_ID" type="xsd:string" use="optional"/>
```

Module

```
<xsd:attribute name="MODULE_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="MODULE_NAME" type="xsd:string" use="optional"/>
<xsd:attribute name="NEXT_MODULE" type="xsd:string" use="optional"/>
<xsd:attribute name="PREVIOUS_MODULE" type="xsd:string" use="optional"/>
<xsd:attribute name="PARAMETER_IDREFS" type="xsd:IDREFS" use="optional"/>
<xsd:attribute name="PRESET_IDREFS" type="xsd:IDREFS" use="optional"/>
```

AuxFile

```
<xsd:attribute name="AUX_FILE_ID" type="xsd:string" use="optional"/>
<xsd:attribute name="PURPOSE" type="xsd:string" use="optional"/>
<xsd:attribute name="LOCTYPE" use="required">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="URN"/>
            <xsd:enumeration value="URL"/>
            <xsd:enumeration value="PURL"/>
            <xsd:enumeration value="HANDLE"/>
            <xsd:enumeration value="DOI"/>
            <xsd:enumeration value="PDI"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
```

Exhibit 2-10. ProcessMD Extension Schema.

2.1.3 Archival Information Classes

The concept of Archival Information Classes (AICs) introduces an additional method for storing and representing metadata. An AIC is a form of AIP that contains information describing characteristics common to a class (group) of AIPs. All AIPs that are members of a particular class, by referencing the ID of that AIC, can inherit its information. The term inherit in this context means that each AIP contains a link to the AIC identifier that allows a user to access the information contained in the AIC directly, as if it were part of the AIP. The internal structure of an AIC is identical to that of an AIP. The only difference between an AIP and an AIC is that the information contained in AIC data files is actually metadata about the characteristics of other AIPs. This allows the metadata to be stored in any MIME type and in any quantities desired. The actual metadata component of an AIC is a description of its data files and thus is really metadata about metadata. It is this recursion that can be confusing at first. As an example, an AIC might contain data files with information about the computing environment used to create a class (group) of AIPs. That information might include documentation, instructional videos, briefings, or the actual software used to create the AIPs. The metadata of that AIC describes the data files containing the computing environment information. Exhibit 2-11 illustrates the AIC concept.

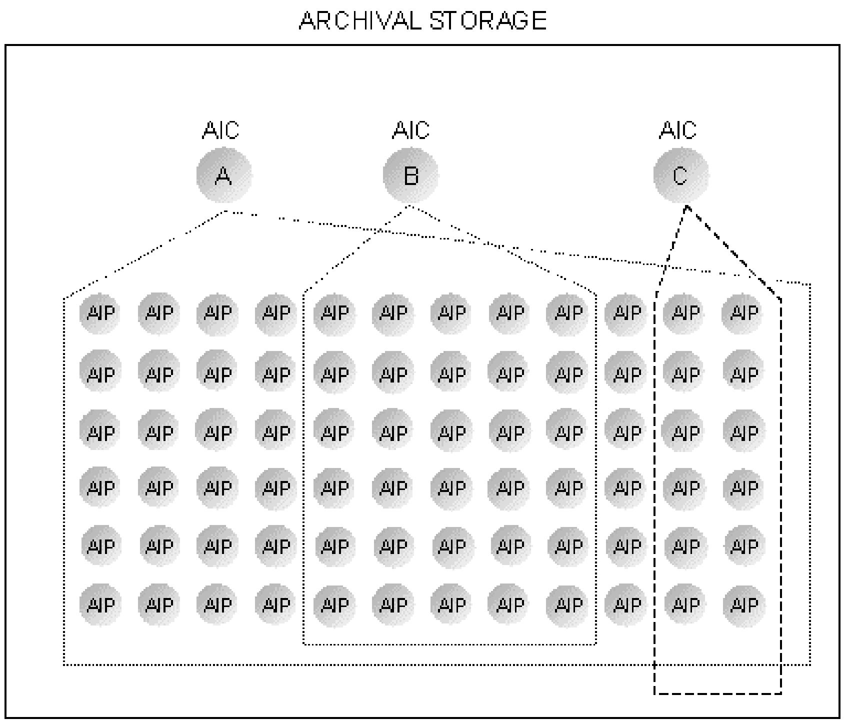


Exhibit 2-11. Archival Information Class Concept.

In Exhibit 2-11 seventy-two (72) AIPs and three AICs reside in Archival Storage. Each AIP represents an archival object such as a motion picture or an audio recording. Although each AIP contains its own unique set of metadata, there may be additional metadata that is common to more than one AIP. As an example, AIC "A" might contain information about the workflow, personnel, and tools employed in the production process for all seventy-two of the AIPs. AIC "B" might contain information describing the specific file format(s) used when digitizing the audio files in thirty (30) of the AIPs. AIC "C" might contain information about the video workstation in which the video files of twelve (12) of the AIPs were created. As can be seen in the Exhibit, AIPs can belong to more than one class and thus inherit metadata from multiple AICs.

The AIP metadata model recommended in this report describes two types of AICs: Representation and Environment. This is not meant to imply that only two types of AICs exist or that these two examples are optimal. The importance of the AIC concept lies in its ability to store a rich variety of metadata for a class of AIPs and to store it only once, rather than replicating it for every AIP. The choice and definition of AICs (in fact the decision to employ AICs at all) must be determined by the LC based upon cost-benefit analysis.

2.1.3.1 Representation AIC

A Representation AIC is an AIP containing information that describes the syntax and semantics of file formats used in a group of AIPs. It describes in detail such things as encoding algorithms, compression schemes, encryption schemes, and conversion routines. The goal of the Representation AIC is to provide sufficient information to support automated rematerialization of a data file or group of data files in a faithful and accurate manner, long after the file formats have become obsolete. Exhibit 2-12 provides an example of the content of a Representation AIC that might be used to help rematerialize a video data file.

Category	Documentation	Instantiation	Data File
Encoder	MPEG-2 Specification: Generic coding of moving pictures and associated audio information ISO/IEC 13818-1:2000 – Systems ISO/IEC 13818-2:2000 – Video ISO/IEC 13818-3:1998 – Audio ISO/IEC 13818-4:1998 – Testing ISO/IEC TR 13818-5:1997 – Software Simulation		 mpeg2_spec1.pdf mpeg2_spec2.pdf mpeg2_spec3.pdf mpeg2_spec4.pdf mpeg2_spec5.pdf
		MPEG-2 Video Encoder source code	mpeg2_encode.src
Decoder	MPEG-2 Specification: system target decoder model ISO/IEC 13818-4:1998/Amd 2:2000		pc_userman.pdf
		MPEG-2 Video Decoder source code	mpeg2_decode.src
Encryption	RSA Encryption Algorithm Specification		rsa_spec.pdf
	RSA BSAFE Software Development Kit Users Manual		rsa_bsafe.pdf
		RSA algorithm source code	rsa_cpp.src
Player	Microsoft Media Player Help File		win_media.htm
	Microsoft Media Player	Windows Media Player source code	wmplayer.src
Conversion	POPcast Video Users Guide		pop_guide.pdf
	POPcast Video Conversion Software	POPcast source code	pop_convert.src

Exhibit 2-12. Example of Representation AIC Data Files.

2.1.3.2 Environment AIC

An Environment AIC is an archival object that contains information about the computing environment in which a group of AIPs can be rematerialized. The data files that comprise an Environment AIC contain information that describes the hardware, system, and software used to rematerialize that class of AIPs.

Environment AIC data files can contain many types of information describing a computing environment, including narrative descriptions, parametric data, databases, documentation, videos, diagrams, schematics, specifications, and/or actual software. The variety and depth of information contained in an Environment AIC may vary from archive to archive depending upon the prerogatives and resources of the Producers and Preservation Service Providers. Exhibit 2-13 provides an example of the data files from an Environment AIC for an audio presentation workstation.

Category	Documentation	Instantiation	Data File
Hardware	Narrative description of the hardware platform used to rematerialize the archival object		pc_desc.txt
+Desktop PC	PC Reference Manual		pc_refman.pdf
	PC Users Manual		pc_userman.pdf
	PC Trouble Guide		pc_trouble.pdf
++16 bit Audio Card	Audio Card Reference Manual		ac_refman.pdf
++Graphics Card	Graphics Card Reference Manual		gc_refman.pdf
Operating System	Narrative description of the operating system software used to rematerialize the archival object		os_desc.txt
+Windows 98	OS Reference Manual		osrefman.pdf
	OS Users Manual		os_userman.pdf
	OS Trouble Guide		ostrouble.pdf
	Window 98 Installation Instructions	Windows 98 Installation CD	windows.pdf windows.exe
++Browser	Netscape 6.0 Reference Manual		net_refman.pdf
++Compiler	Visual Basic 6.0 Reference Manual		vb_refman.pdf
Application	Narrative description of the application software used to rematerialize the archival object		pa_desc.txt
+Presentation App	Presentation App Program Reference Manual		pa_refman.pdf
	Presentation Users Manual		pa_userman.pdf
	Presentation App Program Installation Instructions	Presentation App	pa_install.pdf present.exe

		Installation CD	
+WinAmp	WinAmp Reference Manual		wa_refman.pdf
	WinAmp Installation Instructions	WinAmp Installation CD	wa_install.pdf winamp.exe
+LeadTools	LeadTools Reference Manual		lt_refman.pdf
	WinAmp Installation Instructions	WinAmp Installation CD	lt_install.pdf leadtools.exe

Exhibit 2-13. Example of Environment AIC Data Files.

The information contained in the AIC data files is organized by category and type. The categories are: hardware, operating system software, and application software. The types of information are: documentation and instantiation. The goal of an Environment AIC is to provide sufficient information to support migration of its class of AIPs through changes in technology.

2.2 Data Component

The Data Component of the AIP consists of the essence bitstreams that comprise the archival object. Essence bitstreams are referred to as data files. Data files are categorized by media type. Any number of data files can be contained in an AIP.

2.2.1 Derivative Copies

An AIP may contain more than one derivative copy of a data file for each media type sometimes referred to as versions. Each version serves a separate purpose. The Master version is used for preservation purposes only. It is always the highest quality version. Service versions are used for distribution and presentation. Preview versions are used in discovery and archival object navigation. Examples of the versions, file formats, and specifications for the four media types are listed in Exhibit 2-14. The examples below are not meant to imply recommendations regarding formats and specifications that should be used in an AIP, but rather to indicate the range and variety of file formats and specifications that must be accommodated. It is inevitable that new file formats will arise and old formats will become obsolete. The design of the AIP must be able to adapt to these changes with minimum effort.

Media Type	Version	File Format	Specifications
Text	Master	SGML	ISO/IEC JTC1 SGML standard
	Service	XML	XML 1.0 standard
	Preview	GIF	Graphic Information Format, tonal depth: 8 bit grayscale or indexed color, title page only
Image	Master	TIFF	Tag Image File Format, tonal depth: grayscale: 8 bits per pixel; color: 24 bits per pixel, uncompressed, spatial resolution: 600 dpi
	Service	JPEG	Joint Photographic Experts Group format, tonal depth: grayscale: 8 bits per pixel; color: 24 bits per pixel, compression: 10:1 compression for

			grayscale, 20:1 for color
	Preview	GIF	Graphic Information Format, tonal depth: 8 bit grayscale or indexed color, native LZW system, spatial resolution: 150x150 pixels
Audio	Master	WAV	Uncompressed PCM bitstream, 96 kHz sampling, 24 bit word length
	Service	MP3	Compressed MP3 bitstream, HiFi quality
	Preview	MP3	Compressed MP3 bitstream, LoFi, first 10 seconds only
Video	Master	ITU-R 601	Digital Component Video, uncompressed 4:2:2 ("601") bitstream
	Service	MPEG-2	Moving Pictures Experts Group 2, compressed MPEG-2 bitstreams, 8 Mbits per second
	Preview	GIF	Graphic Information Format, tonal depth: 8 bit grayscale or indexed color, native LZW system, spatial resolution: 150x150 pixels

Exhibit 2-14. Example File Formats.

2.3 Packaging Component

It is proposed that the AIP be packaged as a self-extracting archive. For UNIX-based archives TAR format appears to be the best choice. TAR format software is bundled with the operating systems of most major UNIX vendors, including Sun and HP. For Windows-based archives ProZip appears to be the best choice. Both offer the following features:

- support most standard file types;
- span multiple storage volumes;
- have no file size or archive size restrictions;
- handle large numbers of files;
- provide encryption and loss-less compression; and
- are available at no cost.

Section 3 – Analysis and Recommendations

The purpose of an AIP is to preserve an archival item in digital form for a long time. Given this purpose, it is suggested that the following are desirable qualities of an AIP design:

- Minimizes technology dependence
- Integrates data and metadata
- Accommodates any data format
- Supports both migration and emulation
- Self-contained
- Self-describing
- Self-extracting
- Modular
- Scalable
- Flexible
- Adaptable
- Accessible

Section 3 examines several global issues that affected the preliminary design of the AIP. It analyzes the preliminary design in terms of the desirable qualities listed above. It also provides recommendations regarding the direction of future revisions to the design. Section 3 is divided into four subsections: Metadata, Data, Packaging, and Conclusions and Recommendations.

3.1 Metadata

The analysis begins with a look at the global issues that arose during the preparation of this report. Following the analysis of these issues, the preliminary design of the AIP metadata component, previously outlined in Section 2, is analyzed in greater detail.

DTD vs. Schema

An issue was raised early in the AIP design process regarding the use of an XML schema document type versus XML Document Type Definition (DTD) to define the structure, content, and semantics of metadata component of the AIP. Both are capable of expressing shared vocabularies and allow machines to carry out rules made by people. Based upon a brief review of articles comparing DTDs and schemas, it would appear that DTDs are believed to be best at describing document structures for presentation, and schemas are considered best at describing data types and logical structures for automated processing.

The AIP preliminary design relies on three capabilities that should be supported by the metadata markup language. The first is namespace partitioning that provides “plug and play” flexibility for metadata validation. The second is the availability of automated parsing tools. And the third is strong data typing.

Given the reliance of the AIP design on these three capabilities, it is recommended that XML schema be used to define the structure and content of the metadata component of the AIP. For those interested in a more in-depth comparative analysis of DTD vs. Schema, the following website offers a PowerPoint briefing on the subject: http://www.cen.com/ng-html/xml/schema/DTD-vs-Schema-NASA-TEAS.ppt

METS vs. MPEG-21

Two evolving standards are emerging to define metadata schema(s) for complex multimedia digital objects: METS and MPEG-21. Both are evolving from the metadata requirements imposed by their respective constituencies, and by so doing are taking slightly different approaches. The METS approach focuses on the needs of the digital library community whereas the MPEG-21 approach focuses on the needs of producers, distributors, and consumers of commercial multimedia digital material. The METS approach evolved from the Making of America II project at the University of California, Berkeley and organizes metadata into four primary categories familiar to most digital librarians: descriptive, administrative, file inventory, and structural. The MPEG-21 approach has adopted a single hierarchical view of metadata that for the most part does not distinguish between categories of metadata.

Although a thorough analysis of the advantages and disadvantages of these two standards is beyond the scope of this report, a cursory analysis was performed that resulted in the two key observations.

First, the similarities between the two standards far outweigh the differences. The developers of METS and MPEG-21 have both embraced designs that accommodate multiple schemas. Both have created models that place few if any restrictions on metadata content. Both offer flexible structures for decomposing archival objects into intermediate and terminal objects. And both allow references to external metadata sources.

Second, despite the terminological differences (element and attribute names) between the two approaches, translation between the two should be relatively easy using XSL Stylesheets.

Based upon these observations, and the involvement of the LC in the development of METS, it is recommended that the preliminary design of the AIP metadata component be based upon METS. Appendix A contains a listing of the METS Primary Schema.

Archival Information Class (AIC)

Some metadata describing archival objects can be common to more than one object. As an example, information describing the computing environment in which a group of AIPs were created may be common to many AIPs. The OAIS Reference Model proposes that metadata common to more than one AIP be stored only once and referenced by those AIPs. The OAIS Reference Model suggests that this metadata be stored in the archive in a form of AIP referred to as an Archival Information Class. For all intents and purposes, an AIC is an AIP. It possesses all the characteristics and structure of an AIP, but it contains information about the characteristics of a class (group) of AIPs. The class concept known as inheritance is well known in computer science and is used extensively in object-oriented programming and object-oriented databases.

The use of AICs offers the following advantages:

- the type and amount of information contained in an AIC is discretionary;
- large volumes of metadata can be accommodated easily;
- metadata can be represented in any MIME type, not just text;
- the information contained in an AIC is preserved in the same archive as its AIPs;
- metadata for an entire class of AIPs need only be stored once;
- as a new class is created, a new AIC can be added without significant changes to its AIPs.

Given the advantages cited above, it is recommended that the AIC concept be given serious consideration for implementation at the LC.

3.1.1 METS Primary Schema

The METS Primary Schema is a work in progress. Since its introduction in April 2001, several changes to the METS Primary Schema have been suggested by members of the METS development team. It is anticipated that additional changes will be made over the coming months, resulting in a proposed standard in the Fall of 2001. The Library of Congress Network Development and MARC Standards Office, has volunteered to maintain the emerging METS standard. Appendix A contains a listing of the METS Primary Schema.

3.1.1.1 <dmdSec>

The <dmdSec> element references descriptive metadata. Since the requirements for descriptive metadata vary widely from organization to organization, the <dmdSec> references the <mdSec> sub-element that serves as a placeholder or socket for one or more Extension Schemas containing customized metadata templates._The <mdSec> sub-element implements this reference capability in a sub-element called <mdWrap>. The <mdWrap> sub-element also contains a sub-element called <binData>. The <binData> sub-element is provided to permit the encapsulation of data in Base64 encoding format that might otherwise contain special characters that would cause problems in XML parsing. However, given the importance of XML parsing to AIP processing and the productivity gains derived from using off-the-shelf XML parsing tools, it is recommended that descriptive metadata be stored in an Extension Schema and not in <binData> to avoid potential parsing problems.

3.1.1.2 <amdSec>

Administrative metadata in the <amdSec> element is used to manage and preserve the archival object. <amdSec> contains three sub-elements: <techMD>, <rightsMD>, <sourceMD>, and may soon contain a fourth sub-element called <processMD>. Each of these sub-elements in turn reference the <mdSec> sub-element wherein Extension Schemas may be declared.

The <techMD> sub-element is intended to reference one or more Extension Schemas that describe the technical characteristics of the types and versions of files that are contained in the archival object. The Prototyping Project has chosen to define five (5) Extension Schemas for this purpose: AllFilesMD, TextMD, ImageMD, AudioMD, and VideoMD. (see Section 3.1.2)

The <rightsMD> and <sourceMD> sub-elements within the <amdSec> also reference the <mdSec> sub-element. The Prototyping Project has developed a RightsMD Extension Schema and a SourceMD Extension Schema for these sub-elements respectively. (see Section 3.1.2)

3.1.1.3 <fileGrp>

The <fileGrp> element is used to define the inventory of files contained in an AIP. Its recursiveness allows files to be grouped for purposes of administration. The <file> element contains sub-elements and attributes that define the location and optionally the content of each file in the AIP. The <FContent> sub-element allows the actual content of a binary file to be wrapped in Base64 encoding format and included in the XML. Although this capability may prove useful in the future it is recommended that this capability not be employed in AIPs due to the potential for parsing problems. The <FLocat> sub-element contains information that identifies the location of a data file within the file system. It requires that the type of identifier be specified as URN, URL, PURL, HANDLE, DOI, etc. This raises an interesting implementation issue regarding the relationship between the XML tags and physical names of the files within the AIP itself.

Grouping

The <fileGrp> element of the METS Primary Schema identifies groups of data files contained in the AIP. Each <fileGrp> element describes a separate group of data files. Data files can be organized (grouped) in a variety of ways. They can be organized by media type (i.e., image, text, audio, video) as in <fileGrp>. They can be organized by version. *Note: the term "version" is used within the Prototyping Project to denote how a particular file is used (i.e., Master, Service, Preview).* Files can also be organized by structure (e.g., Side A, Side B) identical to the structure of <div> sub-elements in the <StructMap>. It is recommended that the LC adopt a file grouping convention that mirrors the hierarchical structure of the of <div> sub-elements in the <StructMap> to reduce the complexity of cross-referencing between <fileGrp> elements and <div> elements during XML parsing.

File Naming

Each file in a file group is uniquely identified by a file name and tagged in the instance metadata with an attribute ID. Current prototyping efforts use an attribute ID convention created by appending a one-up number sequence suffix to the archival object identifier. For example, for an archival object identified as "a259", the first file in a file group would have an ID of a259.1.1, the second a259.1.2, the third a259.1.3, etc. Current prototyping efforts use a file naming convention that involves concatenating the identifier of the archival object (e.g.a259) with a suffix that denotes the sub-object type and version of the file (e.g., a259ash= Side A, audio, Service High or a259blm= Side B, label, Master). Although both conventions are workable, it is recommended that the file naming convention adopt the file ID convention to eliminate semantics and reduce XML instance complexity.

AIP Naming Convention

The Library of Congress has developed a Uniform Resource Name (URN) scheme Handle System in conjunction with the Corporation for National Research Initiatives (CNRI). The Handle System is a naming authority (Internet server) that maps uniform logical names to local physical address space.

The AIP preliminary design proposes to use an LC handle as its identifier. The AIP identifier would be composed of the naming authority, the aggregate name, and the item name. In the following example of a handle: "loc.mbrsrs/lp001/a259"

- loc.mbrsrs is the naming authority
- "lp001" is the name of an aggregate or group of recordings.
- "a259" is an identifier assigned to a recording entitled "The Great Ray Charles".

In this handle "loc" stands for Library of Congress and identifies the keeper of this namespace. "mbrsrs" stands for Motion Picture, Broadcasting, and Recorded Sound Division and its Recorded Sound section, the name of the "owner" of the lp001 aggregate and its associated items within the LC. *Note: Although it is recognized that the naming authority is not actually considered part of the handle, it is included here for clarity.*

This approach raises an implementation question: should the individual data files that comprise an AIP have registered handles or should only the AIP itself have a registered handle? If one chooses to access the AIP as an entity and use its self-extracting capabilities to access individual data files, then there is little reason to register separate handles for each data file within an AIP. If, on the other hand, it is desired to access individual data files without the aid of the AIP self-extraction capability, then registering handles for individual data files makes sense. Given the advantages that self-extraction offers, it is recommended that only the AIP itself have a handle and that the physical locations of individual data files be resolved through self-extraction.

3.1.1.4 <structMap>

The <structMap> element of the METS Primary Schema defines a hierarchical structure (or structures) that allows users of the archival object to allow them to navigate through it during presentation. The <structMap> encodes this hierarchy as a nested series of <div> sub-elements. Each <div> carries attribute information specifying what kind of division it is and also may contain multiple file pointer (<fptr>) sub-elements. File pointers specify files (or in some cases, locations within files) that correspond to the portion in the hierarchy represented by the <div>. The <structMap> is a useful and clever mechanism. It permits unlimited nesting and thus accommodates foreseeable presentation needs. In addition, multiple <structMap> elements can be specified to support multiple presentation views.

Labeling

An issue has arisen regarding labeling in the <StructMap>. In the METS Primary Schema, the <div> element defines LABEL as an attribute. As an attribute, the implication is that LABEL is meant to be a simple value and is not intended for complex information. However, current prototyping efforts use the LABEL attribute to contain descriptive metadata about cuts of audio such as title, performer, and duration. This makes parsing of metadata within the LABEL more difficult. There are several ways around this problem. One way is to put this descriptive metadata in two places (i.e., <dmdSec> and <structMap>). Another way is to use the <div> attribute DMD to point to the an Extension Schema for label information. A third way is to change LABEL from an attribute to an element within <div> and include the various types of label metadata in the label as attributes. Given the availability of Extension Schema, it is recommended that the LABEL attribute of <div> contain basic structural information only and that the <div> attribute DMD be used to point to an Extension Schema for descriptive label metadata.

3.1.2 Extension Schemas

This section analyzes the nine (9) proposed Extension Schemas listed in Appendix A.

- GDM
- AllFilesMD
- ImageMD
- TextMD
- AudioMD
- VideoMD
- RightsMD
- SourceMD

- ProcessMD

3.1.2.1 GDM Extension Schema

The GDM Extension Schema contains descriptive metadata about the archival object including bibliographic data, dimensions, origin, creator, etc. The GDM Extension Schema was developed by Rick Beaubien at the University of California, Berkeley. Much of the information is similar in content to fields contained in the M/B/RS metadata database. However, GDM appears to have been intended to capture the key descriptive information about primary objects consisting primarily of text and images. The AIP requires descriptive metadata for describing other media types and sub-objects as well as the primary object. As an example, GDM sub-elements could be used to describe the parts of a record album such as the cover (jacket) or the disc label(s), or the individual cuts on the disc itself. It is recommended that GDM be re-examined to determine if it contains sufficiently generic sub-elements and attributes to accommodate metadata for audio and video media types and intermediate and terminal sub-objects (nodes).

3.1.2.2 AllFilesMD Extension Schema

The AllFilesMD Extension Schema is one of five Extension Schemas developed to describe technical metadata about the files contained in an AIP. AllFilesMD describes the file properties of a file whereas the other four Extension Schemas describe the media properties of a file.

This approach was taken for two reasons. First, it was discovered that a substantial subset of technical information was common to all files regardless of their media type. It was determined that this common information need only be stored once, is less prone to technological obsolescence, and is likely to be accessed more often. Second, it was determined that the technical metadata describing each media type was sufficiently distinct that a separate Extension Schema was needed for each media type. Subsequent research into the technical metadata for each media type revealed the emergence of standards by a variety of national and international organizations. As these standards evolve and mature it is likely that they will supplement or replace the Extension Schemas defined by the Prototyping Project. In fact, the emerging NISO standard for technical image metadata has already been incorporated into the ImageMD Extension Schema.

3.1.2.3 Media Type Extension Schemas (TextMD, ImageMD, AudioMD, and VideoMD)

The AIP design described in Section 2 contains examples of four major media type Extension Schemas: ImageMD, TextMD, AudioMD, and VideoMD. This is not meant to imply that only four media types can be accommodated. On the contrary, the AIP design can easily accommodate other media types by adding new Extension Schemas. As an example, archival items such as maps or software games could be accommodated by simply adding new Extension Schemas that define their technical characteristics (e.g. MapMD, GameMD).

The TextMD and VideoMD Extension Schemas proposed in this report contain limited technical metadata content. This is due primarily to a lack of time to research text and video metadata standards prior to publication of this report. It is recommended that the technical metadata standards for text and video metadata be researched and their Extension Schemas be revised as required.

3.1.2.4 RightsMD Extension Schema

The RightsMD Extension Schema focuses primarily on LC restrictions but does not adequately address the complexity of rights and ownership metadata in a born-digital world. In the future, as born digital material is added to the collections, revisions to the RightsMD Extension Schema should be considered based upon standards like that proposed by the INDECS project or by the RIAA for the inscribing of rights information.

3.1.2.5 SourceMD Extension Schema

The SourceMD Extension Schema contains limited metadata describing text, image, and video sources. This is due primarily to a lack of time to research the types of metadata that might be desired for presentation or required for long-term preservation. It is recommended that additional research be performed to determine the optimum set of metadata for the text, image, and video elements of the SourceMD Extension Schema.

3.1.2.6 ProcessMD Extension

This report embraces the premise that information about the metamorphosis of a digital archival object will become a valuable resource for historical research in the future. A cursory analysis of de facto practices in the computing industry demonstrates a disturbing lack of concern for preserving information about obsolete systems, data formats and the process of migration. Despite the claims that no information is lost in the migration of digital data or emulation of obsolete systems, the rematerialization experience is likely to be different from one generation of an archival object to the next. For this reason it is recommended that the design of the AIP incorporate metadata about the process of digital preservation. Long-term digital preservation involves two different processes: migration and emulation. Migration involves changing old data to accommodate new systems and emulation involves changing new systems to accommodate old data.

Migration

There are two forms of migration: media and format. In media migration, the underlying bitstreams remain unchanged but are moved from storage medium to another. In format migration the bitstreams are transformed from an obsolescent format into a new successor format. Often the migration process involves both forms concurrently.

When is it sensible to migrate content from one format to another? The major drivers in this decision-making process are likely to be the tradeoffs between the value of the data and the time, manpower, and cost for migration. Format migration is accomplished most readily when the old technology and the new technology stand side-by-side, both well known and both still in operational condition. But there is always a risk that an old technology will become obsolete before a project can be mounted to preserve the data. If the Library of Congress intends to migrate its digital collections, the institution must ensure that the knowledge and tools for rematerializing an archival object remain viable until the transformation has occurred and the migration process has been recorded.

Emulation

Although migration is believed by many to be the most practical and cost-efficient method of digital preservation, there are some who believe that emulation too may play a role. Emulation was born in the 1960s and is a technique used to extend the useful life of hardware and software that has become obsolete. It involves building a “virtual” environment within the new technology that emulates the processing characteristics of the environment used to rematerialize the information originally, thus allowing the data to remain unchanged. If at some point it is decided that emulation is a viable option, the need to preserve information about obsolete computer technology becomes even more essential.

This report strongly recommends pursuing the goal of extending METS to include Extension Schemas that record metadata about the preservation process whether it involves migration or emulation. The ProcessMD Extension Schema and the Representation and Environment AICs presented in this report are an initial step in that direction. The intention to make a comprehensive set of schemas for use in an AIP marks a distinction with the intentions of those who first drafted the METS. The hope of the Prototyping Project is that the METS structure will serve both purposes and that the extended reach will be provided by logical and acceptable extensions to the METS structure and acceptance of the AIC concept.

At the time this report was published a discussion was underway among the developers of the METS Primary Schema regarding the inclusion of a separate sub-element in the METS Primary Schema to accommodate metadata about the processes used to migrate or emulate archival objects from one generation to another. Such process information should provide valuable insight into the changes that have occurred from one generation to the next that may have affected the quality or presentation experience of the archival object. It is recommended that a <processMD> sub-element (socket) be added to the METS Primary Schema that can reference, via <mdSec>, an Extension Schema named ProcessMD in which process-related metadata is stored. The ProcessMD schema should also include preservation planning and quality assurance information. In addition, it is recommended that the ProcessMD Extension Schema reference Representation and Environment AICs containing detailed preservation-related metadata about the data file formats and computing environment(s) in which each generation of an AIP can be rematerialized.

Representation AICs

The proposed ProcessMD Extension Schema does not include sufficient metadata about the syntax and semantic information of file formats to support in-depth research about subtle changes that might have occurred during the preservation process. This is due in part to the design of the ProcessMD Extension Schema and in part to the constraints encountered when attempting to capture metadata in textual (XML) form alone. Therefore, it is recommended that the ProcessMD Extension Schema reference an AIC type call Representation that contains a richer variety of syntax and semantic information about data files. (see Section 2.1.3.1)

Environment AICs

The proposed ProcessMD Extension Schema does not include sufficient metadata about the computing environments used to support rematerialization. This is due in part to the design of the ProcessMD Extension Schema and in part to the constraints encountered when attempting to capture metadata in textual (XML) form alone. At first glance, archiving documentation and/or software about obsolete computing environments might appear costly in terms of storage. However, when compared to the cost of trying to recreate the knowledge base of an obsolete technology from lost or discarded printed material, the cost of storing computing environment information as classes of objects pales to insignificance, and a case can be made that it offers an unambiguous safety net for faithful re-materialization in the long-term.

For this reason, it is recommended that the <ProcessMD Extension Schema reference an AIC type called <Environment>. It is also recommended that the <Environment> AIC contain both detailed documentation and actual software for the computing environments in which archival objects were created, stored, and rematerialized. (see Section 2.1.3.2)

3.2 Data

There are several implementation issues regarding AIP data files that can potentially affect the design of the AIP: file partitioning, access time, derivative copies The first is storage. The second is access. All four issues impact the size of the data files contained in the AIP. Analysis of these issues generates questions about the capabilities of the computing environment in which the AIP is implemented. Although the analysis makes several assumptions about these capabilities, it should be noted that the design of the AIP accommodates a wide range of options. It is not the intent of the analysis to dictate the capabilities of the computing environment.

File Partitioning

It is anticipated that the size of some data files in an AIP, and the AIP itself, will be very large (>10GB). Very large files create a unique set of problems in today's computer environments. First and foremost is the problem of storage volume capacity. If the size of a data file approaches or exceeds the capacity of the storage volume on which it resides, problems can occur. One way to get around this problem is to ensure that the Archival Storage subsystem is capable of volume spanning. Another way is to partition very large data files into multiple smaller files. This solves the storage problem but makes managing and presenting the data files much more difficult. A third way is to apply compression to the data files. There are several simple compression techniques such as run-length-encoding (RLE) that can be applied without data loss, but their compression ratios are small. The more robust compression techniques are often source specific and invariably involve some data loss. It is recommended that very large files not be partitioned and that loss-less compression be applied during packaging.

Access Time

The second problem large files impose is access time. Waiting for 10 GB data file to download in real-time is unacceptable to the average user. There are several ways to combat this situation.

The first option is streaming. Streaming allows the user to begin viewing or listening to a file before the entire file has been downloaded. The only direct impact streaming has on the design of the AIP lies in the file formats required. Certain file formats accommodate streaming; others do not. If streaming is desired, service copies of the data files must be stored, or generated "on-the-fly," that accommodate streaming. The following list contains examples of several streaming formats that existed as of this writing:

- WAV – Audio Wave
- AU – Audio Format File
- MPG – Moving Image
- RAV – Real Audio Video
- WMF – Microsoft Media Player
- QTM – Quick Time Movie (Apple)
- AVI – Audio Video Interleaved (Microsoft)
- VBA – Video BAmba (IBM)
- CHA – Cha, Cha, Cha... Audio Player (IBM)
- ABA – Audio BAmba (IBM)

A second option is once again compression. If the user is willing to accept a loss of quality during presentation, the application of compression can significantly reduce the time required to download data files. Most streaming media formats involve compression, with the exception of WAV.

A third option is staging. Staging leads to a fundamental question. Is it a requirement that users access AIPs directly and immediately from the Archival Storage subsystem, or can access be a scheduled event that involves an ordering process with a built-in time delay? The OAIS Reference Model assumes the latter. If the latter is the case, then the size of the files becomes less of an issue.

A fourth option is using dedicated high-speed networks and high-performance workstations for presentation. Many laboratories and research institutions such as the San Diego Supercomputer Center, Lawrence Livermore, and Los Alamos have chosen to solve similar problems by developing dedicated, high-powered networks and workstations to accommodate large data files. The obvious downside is cost.

Given these four options, it is recommended that the AIP be capable of storing Master and Service copies to support delivery of compressed or streaming.

Derivative Copies

A third related issue involves the choice between storing derivative copies of data files in the AIP or creating them "on-the-fly." Storing derivative copies eliminates the need to create copies "on-the-fly" for presentation or distribution purposes. However, it demands more archival storage space. Clearly, storing the Master copy of each data file in the AIP goes without question. The rationale for storing lower quality copies is driven by distribution and presentation requirements. There are two types of lower quality copies: Service and Preview. A Service copy is a smaller scale version of the Master that maintains acceptable presentation quality. A Preview version is an iconic representation of the Master that is used for identification purposes only. At present, creating Service copies of Master audio, video, and image files "on-the-fly" requires considerable processing power. However, creating Preview copies from Service copies requires considerably less processing power. It is recommended that the AIP contain the Master and one Service copy of each data file and that Preview copies be created "on-the-fly".

Error Detection and Correction

Checksums are a useful and efficient means of detecting errors in a data file, but they do nothing to correct errors once they have occurred. In those cases where the digital form of an archival object is the only viable version that exists, it is important that there be a means to correct digital errors should they occur. There are two ways to accomplish this. The first, and perhaps most straightforward, way is to save multiple copies of the digital archival object (backup). It is recommended that backup copies of all AIPs be created as a matter of policy.

A supplemental approach to consider is error-detection-and-correction (EDAC) codes. Techniques such as Reed-Solomon Codes cross-interleave information with parity data and are able to detect and correct both single-bit and burst errors of variable length. Reed-Solomon is particularly popular because it minimizes the parity overhead and is simple to decode. Reed-Solomon decoding software is readily available at no cost and operates in real time on today's computing platforms. For this reason it should cause no problems in playback. It is recommended that Reed-Solomon coding be applied to the Master audio and video data files in the AIP.

3.3 Packaging

Packaging Information is that information that either actually or logically binds or relates the components of the AIP into an identifiable entity for the purpose of transmission or storage. Packaging offers a number of benefits to the design of an AIP. First, treating a set of files as one contiguous file simplifies transmission and storage. Second, it allows one-pass encryption, compression, and watermarking.

There are a variety of packaging alternatives, such as AIPS++ used to package astronomical files, TAR that is used extensively in Unix environments, and ZIP products used in Windows environments. The most attractive packaging construct is the self-extracting archive. Self-extracting archives have been around for decades. They were originally developed to facilitate backup and recovery but are most well known in their compression role in products such as WinZip, ProZip, and DynaZip. Self-extracting archives offer several distinct advantages. First, they are ubiquitous and inexpensive or free. Second, they provide flexibility and infrastructure independence. Third, they hide unique file format characteristics. For these reasons, it is recommended that self-extracting archive techniques be used to package AIPs for transmission to and from the Archival Storage system.

3.4 Conclusions and Recommendations

Section 3.4 summarizes the conclusions and recommendations of this report.

3.4.1 Conclusions

The proposed AIP preliminary design is modular, infrastructure-independent, flexible, and adaptable. It is consistent with the OAIS Reference Model and complies with the emerging Metadata Encoding and Transfer Standard. It incorporates most of the fields contained in the current Prototyping Project metadata database and accommodates the future inclusion of emerging metadata standards such as MPEG-21.

The AIP metadata is structured using the XML schema document type that facilitates strong data typing. It provides for multiple namespaces (Extension Schema) to support organization-unique metadata requirements. The AIP preliminary design introduces the concept of AIP classes (AICs) allowing for the inheritance of metadata by multiple AIPs. It addresses the issues of long-term preservation of digital information by employing a ProcessMD Extension Schema and Representation and Environment AICs to incorporate the metadata needed to document migration of AIPs through changes in technology. Self-extracting archive techniques are proposed to encapsulate the AIP, allowing it to be treated as a single entity and thereby improving manageability and facilitating infrastructure independence.

3.4.2 Recommendations

The following recommendations are made regarding AIP preliminary design follow-on activity.

- It is recommended that Extension Schemas be refined based on emerging metadata standards.
- It is recommended that a sample set of AIPs be handcrafted for test purposes.
- It is recommended that the LC Digital Preservation Committee develop a plan for AIP testing.
- It is recommended that AIP testing be performed and the results of the testing be used to refine the AIP design.
- It is recommended that the refined AIP design be submitted to the Digital Library Federation for consideration by the group developing the METS schema.
- It is recommended that an XSL stylesheet be developed to translate the METS Primary Schema into an MPEG-21 compliant schema.
- It is recommended that the Prototyping Project metadata database be updated to reflect the AIP design.

Once again the author wishes to thank those members of LC staff that contributed to this report and in particular Carl Fleischhauer, Dick Thaxter, and Morgan Cundiff, without whom this report could not have been prepared.

Appendix A METS Primary Schema.


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Jerome McDonough (private) -->
<!-- METS: Metadata Encoding and Transmission Standard -->
<!-- Prepared for the Digital Library Federation by Jerome McDonough, New York University.-->
<!-- April 23, 2001 -->
<!-- Version 1.0 (alpha) -->
<!-- Change History -->
<!-- April 23, 2001: Alpha Draft completed -->
<xsd:schema targetNamespace="http://www.loc.gov/METS/" xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns="http://www.loc.gov/METS/" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="mets" type="metsType">
        <xsd:annotation>
            <xsd:documentation>METS: Metadata Encoding and Transmission Standard.
            METS is intended to provide a standardized XML format for transmission
            of complex digital library objects between systems.  As such, it can be seen
            as filling a role similar to that defined for the Submission Information Package
            (SIP) and Dissemination Information Package (DIP) in the Reference Model
            for an Open Archival Information System.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="dmdSec" type="mdSecType">
        <xsd:annotation>
            <xsd:documentation>dmdSec: Description Metadata Section.
            This section records all of the descriptive metadata for all subobjects in the METS object.
            Metadata can be either included in the METS hub document (mdWrap) or
            referenced via an identifier/locator (mdRef), a la Warwick Framework.  Multiple
            mdRef and mdWrap elements are allowed so that descriptive metadata
            can be recorded for each separate subobject within the METS object.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="amdSec" type="amdSecType">
        <xsd:annotation>
            <xsd:documentation>amdSec: Administrative Metadata Section.
            This section records all of the administrative metadata for all subobjects in the METS object,
            and is divided into three subsections: techMD (technical metadata), rightsMD
            (intellectual property rights metadata), and sourceMD (provenance metadata).
            Each of these subsections follows the mdSecType model, so that they can
            either include metadata within the METS hub document (mdWrap) or
            reference it via an identifier/locator (mdRef).  Multiple
            mdRef and mdWrap elements are allowed so that administrative metadata
            can be recorded for each separate subobject within the METS object.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="fileGrp" type="fileGrpType">
        <xsd:annotation>
            <xsd:documentation>fileGrp: File Group.
            File Groups record information regarding all of the data files comprising
            the digital object, including both master files and derivatives.  File Groups
            may both repeat and nest to provide an organizing framework for data files.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="structMap" type="structMapType">
        <xsd:annotation>
            <xsd:documentation>structMap: Structural Map.
            The structural map is the heart of a METS document, defining the
            hierarchical arrangement of a primary source document which has
            been digitized.  This hierarchy is encoded as a tree of 'div' elements.
            Any given 'div' can point to another METS document via the 'mptr'
            element, or to a single file, to a group of files, or to segments of individual
            files or groups of files through the 'fptr' and subsidiary elements.
```

```xml
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="div" type="divType">
        <xsd:annotation>
            <xsd:documentation>div: Division of a structural map.
                Nested div elements define the hierarchical structure of the digital library object.
                Div elements are linked to the content corresponding to that div by subsidiary mptr and
                fptr elements.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="metsType">
        <xsd:annotation>
            <xsd:documentation>mets Complex Type.
            A METS document consists of four possible subsidiary sections:
            dmdSec (descriptive metadata section), amdSec (administrative
            metadata section), fileGrp (file inventory group), and structMap
            (structural map).  It also has seven possible attributes:
            1. ID (an XML ID);
            2. OBJID: a primary identifier assigned to the original source
            document;
            3. LABEL: a title/text string identifying the document for
            users;
            4. TYPE: a type for the object, e.g., book, journal, stereograph, etc.;
            5. CREATEDATE: the date the METS object was created;
            6. LASTMODDATE: the date the METS object was last modified; and
            7. PROFILE: the registered profile to which  this METS document conforms.
            METS registry information is available from the Library of Congress at
            http://www.loc.gov/mets.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="dmdSec" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="amdSec" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="fileGrp" maxOccurs="unbounded"/>
            <xsd:element ref="structMap" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
        <xsd:attribute name="OBJID" type="xsd:string" use="optional"/>
        <xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
        <xsd:attribute name="TYPE" type="xsd:string" use="optional"/>
        <xsd:attribute name="CREATEDATE" type="xsd:date" use="optional"/>
        <xsd:attribute name="LASTMODDATE" type="xsd:date" use="optional"/>
        <xsd:attribute name="PROFILE" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="amdSecType">
        <xsd:annotation>
            <xsd:documentation>amdSecType: Complex Type for Administrative Metadata.
            The administrative metadata section consists of three possible subsidiary
            sections: techMD (technical metadata for text/image/audio/video files),
            rightsMD (intellectual property rights metadata), and sourceMD (source
            metadata i.e. provenance).  amdSecType
            has a single attribute, ID (XML ID).</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="techMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation>techMD: technical metadata.
                        The techMD element provides a wrapper around a generic metadata section,
                        which should contain technical metadata regarding a file or files.  It has a single
                        attribute, ID, which file/fileGrp elements can use to reference the technical
                        metadata that applies to them.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="rightsMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
```

```
                <xsd:annotation>
                    <xsd:documentation>rightsMD: intellectual property rights metadata.
                        The rightsMD element provides a wrapper around a generic metadata section,
                        which should contain IP rights metadata.  It has a single attribute, ID, which
                        file/fileGrp/div elements can use to reference IP Rights metadata that applies to them.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="sourceMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation>sourceMD: source metadata.
                        The sourceMD element provides a wrapper around a  generic metadata section
                        which should contain source/provenance information.  It has a single attribute,
                        ID, which file/fileGrp elements can use to reference the source metadata which
                        applies to them.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="fileGrpType">
        <xsd:annotation>
            <xsd:documentation>fileGrpType: Complex Type for File Groups.
                The file group is used to cluster all of the digital files composing a digital
                library object in a hierarchical arrangement (fileGrp is recursively defined
                to enable the creation of the hierarchy).  Any file group may contain zero or
                more file elements.  File elements in turn can contain a FLocat element
                (a pointer to a file containing content for this object) and/or a FContent
                element (the contents of the file Base64 encoded).  a fileGrp element
                may have the following attributes:
                1. ID: an XML ID for the element
                2. VERSDATE: date this version/fileGrp of the digital object was created.
                3. ADMID: IDREFs to administrative metadata sections in the METS document
                that correspond with all files in this file group.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:choice>
            <xsd:element ref="fileGrp" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="file" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation>file: File element.
                        The file element provides access to content files for a METS document.
                        It has the following attributes:
                        1. ID: an XML ID for the element;
                        2. MIMETYPE: the MIME type for the file;
                        3. SEQ: an integer indicating the sequence of this file relative to the others in its file group;
                        4. SIZE: the size of the file in bytes;
                        5. CREATED: the date of creation for the file;
                        6. OWNERID: a primary identifier assigned to the file by its owner;
                        7. ADMID: IDREFS to administrative metadata sections in the METS document
                        that correspond with this file; and
                        8. GROUPID: an identifier that establishes a correspondence between this file
                        and files in other file groups.  Typically, this will be used to associate a master
                        file in one file group with derivative files in other file groups.
                        A file element may contain an FLocat element, which provides a pointer to
                        a content file, and/or an FContent element, which wraps an encoded version
                        of the file.
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:all>
                        <xsd:element name="FLocat" minOccurs="0">
                            <xsd:annotation>
                                <xsd:documentation>FLocat: File Location.
                                    The FLocat element provides a pointer to the location of a content file.
```

```
                        It has three attributes:
                        1. ID (an XML ID);
                        2. LOCTYPE: the type of locator     contained in the FLocat element; and
                        3. OTHERLOCTYPE: a string to indicate an alternative LOCTYPE if
                        the LOCTYPE attribute itself has a value of "OTHER."
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                            <xsd:attributeGroup ref="LOCATION"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="FContent" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>FContent: file content.
                        The FContent element is used to deliver a content file for a METS
                        document within the METS file itself.  The content file must be Base 64
                        encoded, and contained within the FContent wrapper element.  The
                        FContent element ahs the following attributes:
                        1. ID (an XML ID); and
                        2. CHECKSUM: an MD5 checksum value for the included file.
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:binary">
                            <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                            <xsd:attribute name="CHECKSUM" type="xsd:string" use="optional"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
        <xsd:attribute name="ID" type="xsd:ID" use="required"/>
        <xsd:attribute name="MIMETYPE" type="xsd:string" use="optional"/>
        <xsd:attribute name="SEQ" type="xsd:int" use="optional"/>
        <xsd:attribute name="SIZE" type="xsd:long" use="optional"/>
        <xsd:attribute name="CREATED" type="xsd:date" use="optional"/>
        <xsd:attribute name="OWNERID" type="xsd:string" use="optional"/>
        <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional"/>
        <xsd:attribute name="GROUPID" type="xsd:string" use="optional"/>
    </xsd:complexType>
</xsd:element>
</xsd:choice>
<xsd:attribute name="ID" type="xsd:ID" use="optional"/>
<xsd:attribute name="VERSDATE" type="xsd:date" use="optional"/>
<xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional"/>
</xsd:complexType>
<xsd:complexType name="structMapType">
    <xsd:annotation>
        <xsd:documentation>structMap Complex Type
        The structural map (structMap) outlines a hierarchical structure for the
        original object being encoded, using a series of nested div elements.
        The structMap element has the following attributes:
        1. ID: an XML ID for the element;
        2. TYPE: the type of structural map provided.  Typical values will be
        "PHYSICAL" for a map which describes the physical composition of
        the original work (a series with individual monographs with pages) and
        "LOGICAL" for one which describes the intellectual structure of the work
        (a monograph with TOC, forward, chapters, index., etc.);
        3. LABEL: a string to describe the structMap to users.  This is primarily
        useful where more than one subject is provided for a single object
```

```
            (e.g., both logical and physical structMap).</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="div" type="divType">
                <xsd:annotation>
                    <xsd:documentation>div: Division.
                        The METS standard represents a document structurally as a series of nested
                        div elements, that is, as a hierarchy (e.g., a book, which is composed of chapters,
                        which is composed of subchapters, which is composed of text).  Every div node
                        in the structural map hierarchy may be connected (via subsidiary mptr or fptr
                        elements) to content files which represent that div's portion of the whole document.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
        <xsd:attribute name="TYPE" type="xsd:string" use="optional"/>
        <xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="divType">
        <xsd:annotation>
            <xsd:documentation>Div Complex Type
                    The METS standard represents a document structurally as a series of nested
                    div elements, that is, as a hierarchy (e.g., a book, which is composed of chapters,
                    which is composed of subchapters, which is composed of text).  Every div node
                    in the structural map hierarchy may be connected (via subsidiary mptr or fptr
                    elements) to content files which represent that div's portion of the whole document.
                    The div element has the following attributes:
                    1. ID (an XML ID);
                    2. ORDER: an integer representation of this div's order among its siblings
                    (e.g., its page number);
                    3. ORDERLABEL: a string representation of this div's order among its siblings (e.g. "Page xii)";
                    4. LABEL: a string to describe this div to an end user viewing the document, as per
                    a table of contents entry; and
                    5. DMD:  a set of IDREFs to descriptive metadata sections within this METS document
                    applicable to this div.
                    6. ADMID: a set of IDREFS to administrative metadata sections within this METS document
                    applicable to this div.
                </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="mptr" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation>mptr: METS Pointer.
                        The mptr element allows a div to be associated with a separate METS document
                        containing the content corresponding with that div, rather than pointing to an
                        internal file or file group.  A typical instance of this would be the case of a METS
                        document for a journal run, with a div elements for each individual journal issue.
                        The div elements for the issues might point to separate METS documents for each
                        issue, rather than having files and file groups for every issue encoded in one
                        document.  The mptr element may have the following attributes:
                        1. ID: an XML ID for this element;
                        2. LOCTYPE: the type of locator      contained in the FLocat element; and
                        3. OTHERLOCTYPE: a string to indicate an alternative LOCTYPE if
                        the LOCTYPE attribute itself has a value of "OTHER."
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                            <xsd:attributeGroup ref="LOCATION"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
```

```
        <xsd:element name="fptr" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>fptr: File Pointer.
                    The fptr element associates a div element with content files that represent that div.
                    It can either point to a file directly itself, via the FILEID attribute, or it can do more
                    complex links to content via the subsidiary par and seq elements.  The fptr
                    element can have the following attributes:
                    1. ID: an XML ID for this element; and
                    2. FILEID: an IDREF to a file element which corresponds with the div containing
                    this ftpr.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:complexType>
                <xsd:choice>
                    <xsd:element name="par" minOccurs="0">
                        <xsd:annotation>
                            <xsd:documentation>par: Parallel  files.
                                The par element should used to link a div to a set of content files when
                                those files should be played back in unison to display the content to the
                                user.  Individual area subelements within the par element provide the links
                                to the files or portions thereof.  Par has the following attributes:
                                1. ID: an XML ID for this element.
                            </xsd:documentation>
                        </xsd:annotation>
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="area" type="areaType" maxOccurs="unbounded"/>
                            </xsd:sequence>
                            <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="seq" minOccurs="0">
                        <xsd:annotation>
                            <xsd:documentation>seq: Sequence of files.
                                The seq element should be used to link a div to one or more content files
                                when those files should be played sequentially to display content to a user,
                                or where there is only one file to link, but the link from div to file requires
                                identifying a subcomponent of a file.  Seq has the following attributes:
                                1. ID: an XML ID for this element.
                            </xsd:documentation>
                        </xsd:annotation>
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="area" type="areaType" maxOccurs="unbounded"/>
                            </xsd:sequence>
                            <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="area" type="areaType" minOccurs="0"/>
                </xsd:choice>
                <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                <xsd:attribute name="FILEID" type="xsd:IDREF" use="optional"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element ref="div" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
    <xsd:attribute name="ORDER" type="xsd:integer" use="optional"/>
    <xsd:attribute name="ORDERLABEL" type="xsd:string" use="optional"/>
    <xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
    <xsd:attribute name="DMD" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional"/>
</xsd:complexType>
<xsd:complexType name="areaType">
    <xsd:annotation>
        <xsd:documentation>areaType: Complex Type for Area linking.
```

```xml
            The area element provides for more sophisticated linking between a div
            element and content files representing that div, be they text, image,
            audio, or video files.  An area element can link a div to a point
            within a file, to a one-dimension segment of a file (e.g., text screen,
            image line, audio/video clip), or a two-dimensional section of a file
            (e.g, subsection of an image, or a subsection of the  video display
            of a video file.  The area element has no content, and the following
            attributes:
            1. ID: an XML ID;
            2. FILEID: an IDREF to the file element being pointed to by the div;
            3. SHAPE: a text string defining the shape of a two-dimensional
            area being referenced in a link file;
            4. COORDS: a text string representing a set of visual coordinates
            within an image (still image or video frame).  The COORDS and
            SHAPE attributes should be used as in HTML 4;
            5. BEGIN: a beginning location in a referenced file;
            6. END: an ending location in a referenced file;
            7. BETYPE: the syntax used in specifying the BEGIN and END
            attributes (byte offset, IDREF value, or SMPTE time code);
            8. EXTENT: the duraction of the segment; and
            9. EXTTYPE: the syntax used in specifying the extent (byte length
            or SMPTE time value).
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
    <xsd:attribute name="FILEID" type="xsd:IDREF" use="required"/>
    <xsd:attribute name="SHAPE" use="optional">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="RECT"/>
                <xsd:enumeration value="CIRCLE"/>
                <xsd:enumeration value="POLY"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="COORDS" type="xsd:string" use="optional"/>
    <xsd:attribute name="BEGIN" type="xsd:string" use="optional"/>
    <xsd:attribute name="END" type="xsd:string" use="optional"/>
    <xsd:attribute name="BETYPE" use="optional">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="BYTE"/>
                <xsd:enumeration value="IDREF"/>
                <xsd:enumeration value="SMIL"/>
                <xsd:enumeration value="MIDI"/>
                <xsd:enumeration value="SMPTE-25"/>
                <xsd:enumeration value="SMPTE-24"/>
                <xsd:enumeration value="SMPTE-DF30"/>
                <xsd:enumeration value="SMPTE-NDF30"/>
                <xsd:enumeration value="SMPTE-DF29.97"/>
                <xsd:enumeration value="SMPTE-NDF29.97"/>
                <xsd:enumeration value="TIME"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="EXTENT" type="xsd:string" use="optional"/>
    <xsd:attribute name="EXTTYPE" use="optional">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="BYTE"/>
                <xsd:enumeration value="SMIL"/>
                <xsd:enumeration value="MIDI"/>
                <xsd:enumeration value="SMPTE-25"/>
                <xsd:enumeration value="SMPTE-24"/>
                <xsd:enumeration value="SMPTE-DF30"/>
                <xsd:enumeration value="SMPTE-NDF30"/>
```

```
                <xsd:enumeration value="SMPTE-DF29.97"/>
                <xsd:enumeration value="SMPTE-NDF29.97"/>
                <xsd:enumeration value="TIME"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional"/>
</xsd:complexType>
<xsd:complexType name="mdSecType">
    <xsd:annotation>
        <xsd:documentation>mdSec (metadata section) Complex Type
        A generic framework for pointing to/including metadata within a METS document,
        a la Warwick Framework.  An mdSec element may have the following attributes:
        1. ID: an XML ID for this element.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
        <xsd:element name="mdRef" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>mdRef: metadata reference.
                    The mdRef element is a generic element used throughout the METS schema
                    to provide a pointer to metadata which resides outside the METS document.
                    It has the following attributes:
                    1. ID: an XML ID;
                    2. LOCTYPE: the type of locator contained in the body of the element;
                    3. OTHERLOCTYPE: a string indicating an alternative LOCTYPE when the LOCTYPE
                    attribute value is set to "OTHER.";
                    4. MIMETYPE: the MIME type for the metadata being pointed at;
                    5. MDType: the type of metadata being pointed at (e.g., MARC, EAD, etc.);
                    6. OTHERMDTYPE: a string indicating an alternative MDTYPE when the MDTYPE
                    attribute value is set to "OTHER.";
                    7. LABEL: a label to display to the viewer of the METS document identifying the metadata;
                    8. XPTR: an xptr to a location within the file pointed to by the mdRef element, if applicable.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                        <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                        <xsd:attributeGroup ref="LOCATION"/>
                        <xsd:attribute name="MIMETYPE" type="xsd:string" use="optional"/>
                        <xsd:attributeGroup ref="METADATA"/>
                        <xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
                        <xsd:attribute name="XPTR" type="xsd:string" use="optional"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="mdWrap" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>mdWrap: metadata wrapper.
                    The mdWrap element is a generic element used throughout the METS schema  to allow
                    the encoder to place arbitrary metadata conforming to other standards/schema within a
                    METS document.  The included metadata can either be encoded in XML, in which case
                    it may be placed directly within the mdWrap element, or it can be Base64 encoded, and
                    placed within a subsidiary binData element.  The mdWrap element can have the following
                    attributes:
                    1. ID: an XML ID for this element;
                    2. MIMETYPE: the MIME type for the metadata contained in the element;
                    3. MDType: the type of metadata contained (e.g., MARC, EAD, etc.);
                    4. OTHERMDTYPE: a string indicating an alternative MDTYPE when the MDTYPE
                    attribute value is set to "OTHER.";
                    5. LABEL: a label to display to the viewer of the METS document identifying the metadata.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:complexType>
                <xsd:choice>
```

```xml
                        <xsd:element name="binData" type="xsd:binary" minOccurs="0">
                            <xsd:annotation>
                                <xsd:documentation>A wrapper to contain Base64 encoded metadata.
                                    </xsd:documentation>
                            </xsd:annotation>
                        </xsd:element>
                        <xsd:any namespace="##any"/>
                    </xsd:choice>
                    <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
                    <xsd:attribute name="MIMETYPE" type="xsd:string" use="optional"/>
                    <xsd:attributeGroup ref="METADATA"/>
                    <xsd:attribute name="LABEL" type="xsd:string" use="optional"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
        <xsd:attribute name="ID" type="xsd:ID" use="required"/>
    </xsd:complexType>
    <xsd:attributeGroup name="METADATA">
        <xsd:attribute name="MDTYPE" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="MARC"/>
                    <xsd:enumeration value="EAD"/>
                    <xsd:enumeration value="VRA"/>
                    <xsd:enumeration value="DC"/>
                    <xsd:enumeration value="NISOIMG"/>
                    <xsd:enumeration value="LC-AV"/>
                    <xsd:enumeration value="VRA"/>
                    <xsd:enumeration value="TEIHDR"/>
                    <xsd:enumeration value="DDI"/>
                    <xsd:enumeration value="FGDC"/>
                    <xsd:enumeration value="OTHER"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="OTHERMDTYPE" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="LOCATION">
        <xsd:attribute name="LOCTYPE" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="URN"/>
                    <xsd:enumeration value="URL"/>
                    <xsd:enumeration value="PURL"/>
                    <xsd:enumeration value="HANDLE"/>
                    <xsd:enumeration value="DOI"/>
                    <xsd:enumeration value="OTHER"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="OTHERLOCTYPE" type="xsd:string" use="optional"/>
    </xsd:attributeGroup>
</xsd:schema>
```

Appendix B Survey of Related Projects

Background

Appendix B surveys the background of five projects that have influenced this report, including the Open Archival Information System Reference Model Project (OAIS); the Making of America II Project (MOA2); the ARCHIVES II NARA Project; the Universal Preservation Format Project; and the MPEG-21 Multimedia Framework.

The purpose of the survey is to acknowledge the source of many of the concepts and ideas expressed in the report and to provide a context for the reader. Acknowledgements are marked in italics.

B.1 Open Archival Information System Reference Model Project (OAIS)

Among the more notable and widely accepted developments in long-term digital preservation is the Open Archival Information System (OAIS) Reference Model Project. The major purpose of the OAIS Reference Model Project is to facilitate a wider understanding of what is required to preserve and access digital information for the long term. One of the concepts of the OAIS Reference Model is the Archival Information Package (AIP). *This report bases the AIP architecture on the OAIS AIP concept.*

B.1.1 OAIS Functional Model.

In its functional form, the OAIS consists of 5 entities or subsystems. Exhibit B-1 illustrates the OAIS functional model.

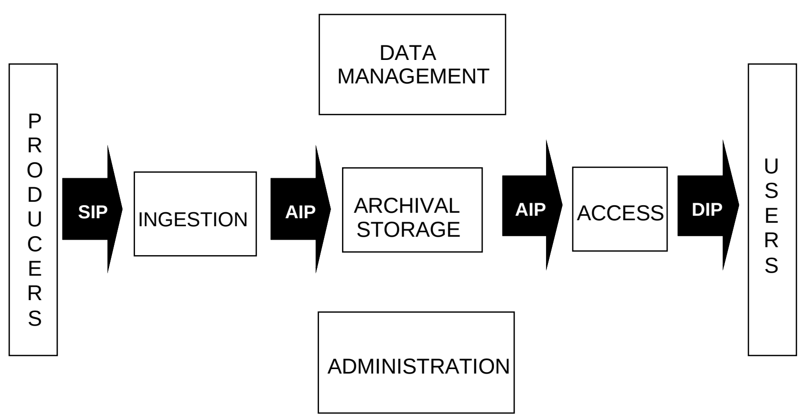


Exhibit B-1. The OAIS Functional Model.

Each of the five subsystems performs a specific role in the preservation process. Ingestion accepts digital information from Producers and prepares it for storage and management within the archive. Archival Storage stores and maintains digital information. Data Management supports the management and discovery of digital information within the archive. Administration coordinates the overall operation of the archive. Access retrieves digital information from the archive and makes it available to Consumers.

B.1.2 Information Packages

The form in which digital information moves between subsystems is referred to as Information Packages. There are three types of Information Packages: Submission Information Packages (SIP); Archival Information Packages (AIP); and Dissemination Information Packages (DIP). SIPs are the form in which Producers transmit digital information to the Ingestion subsystem. AIPs are the form in which the Ingestion subsystem transmits digital information to the Archival Storage subsystem and in which the Archival Storage subsystem transmits digital information to the Access subsystem. DIPs are the form in which the Access subsystem transmits digital information to the Users. An Information Package consists of three major components: Data, Metadata, and Packaging Information. Data is the digitized essence bitstream s that comprise the archival object. Metadata is information that describes the Data. Packaging Information binds the Data and Metadata files together as a single contiguous file.

B.1.3 Archival Information Package

An Archival Information Package (AIP) is a conceptual container of information. Associated with the AIP are four types of information: Content Information, Preservation Description Information (PDI), Descriptive Information, and Packaging Information. Content Information is the item to be preserved and re-materialized such as a digitized phonograph record or a motion picture. Preservation Description Information is the information needed to accurately describe the Content Information and provide an understanding of the environment in which the Content Information was created. Descriptive Information is the information that allows users to discover the Content Information in the archive. Packaging information is the information that binds or encapsulates the Content Information and Preservation Description Information for transmission between subsystems. Exhibit B-2 illustrates the taxonomy of an AIP.

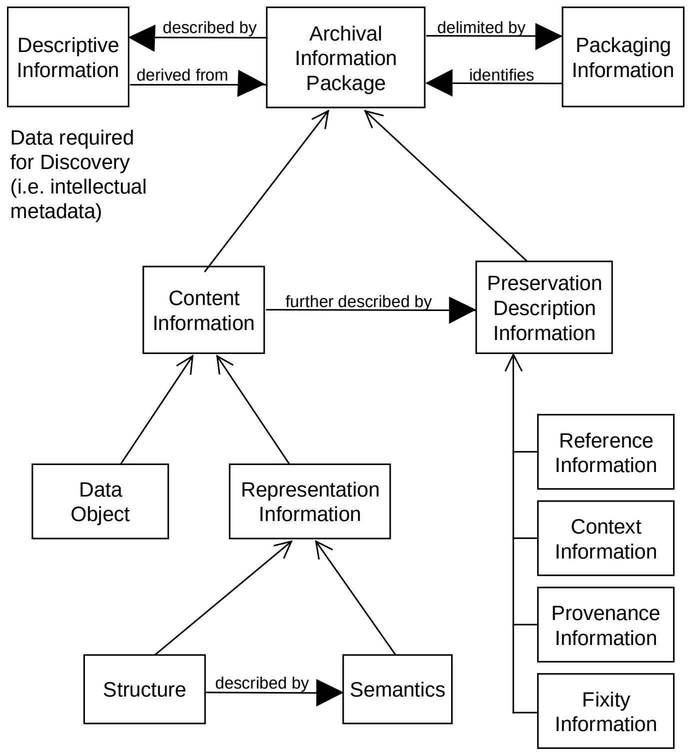


Exhibit B-2. The Archival Information Package (AIP) Taxonomy.

This report leans heavily on the terminology and description of the OAIS Reference Model. The OAIS Reference Model arose from NASA's need to preserve vast collections of digital imagery and telemetry data. It has become the suggested standard of the Consultative Committee for Space Data Systems and has been implemented successfully by NASA in the EOS-DIS digital archive. It has been proposed as an International Standards Organization standard. The OAIS Reference Model has been adopted by the Cedars Project, NEDLIB, OCLC/RLG, and is being implemented in a joint research project conducted by the San Diego Supercomputer Center, sponsored by the National Archives and Records Administration. In addition, the National Research Council recommended, (in a report entitled LC21: A Digital Strategy for the Library of Congress), that the Library of Congress coordinate its efforts to develop a digital repository system with those organizations that are using the OAIS Reference Model.

B.2 Making of America II Project

The Making of America II (MOA2) Project is a University of California at Berkeley effort, in association with the Digital Library Federation, to create a proposed digital library object standard by encoding defined descriptive, administrative and structural metadata, along with the primary content, inside a digital library object. The cornerstone of the MOA II effort is an XML DTD that defines the digital object's elements and encoding. Appendix A contains a listing of the MOA2 XML DTD. The project has also developed a relational database that allows a library to capture the metadata, a program that reads the database and automatically creates the XML encoded digital objects, a repository manager that provides distributed network access to the objects (via RMI), and a viewer that displays MOA II objects from the repository.

Much of the terminology used in this report for the metadata component of the AIP is derived from the MOA2 DTD. In particular, the terms descriptive, administrative, and structural metadata were borrowed from the MOA2 DTD. As previously mentioned, the concept of Primary and Extension Schemas originated from the emerging METS development effort.

B.3 Archives II NARA Project

The Archives II NARA Project is a joint effort of the National Archives and Records Administration and the San Diego SuperComputer Center to develop prototypes for collection-based persistent archive infrastructure. An initial prototype defined a migratable information architecture and a scalable data archive. The key concept developed in the prototype was the provision of separate information descriptions for the digital object structure, organization, and presentation interface. This allowed the:

- Separation of data archiving from data accessioning
- Separation of management of the organization from storage of digital objects
- Separation of presentation and querying from management

The distribution of functionality into separate components was made possible through the use of a data handling system (the San Diego Supercomputer Center Storage Resource Broker) that connected data flows across all of the systems. The information model used to describe the context of each component was based upon a semi-structured representation through use of eXtensible Markup Language Document Type Definitions.

This report borrowed the concept of separate information descriptions for the digital object structure, organization, and presentation interface. It also embraces the concept of a Storage Resource Broker as a means of decoupling the AIP from its physical storage location.

B.4 Universal Preservation Format Project

The Universal Preservation Format (UPF) Project is a research effort sponsored by the WGBH Educational Foundation to develop a storage system for electronically generated content. UPF requires a storage technology that is “self-describing.” A “self-describing” storage technology is one where the structure of the storage need not be known. The structure is disclosed internally in the storage. Access to the storage system is described. Once access is obtained, all information for the retrieval of stored content is described as well as the structure of the content. This “self-description” protects against technological obsolescence.

The “self-describing” storage technology uses a “wrapper” concept to hold the content (defined as metadata plus essence bitstream s). Physically connected to the content is a description, in human readable form, of the physical format housing the storage format. This description describes the construction of a “reader” for the physical format, thereby insuring recovery of the content, even in the absence of the original recording mechanism (i.e. physical storage device).

This report borrowed the “self-describing” concept and implements it in the form of Environment and Semantic AICs containing information about the mechanisms required to re-materialize classes of archival objects. It also borrows the “wrapper” concept and implements it in the form of a self-extracting archive for the AIP.

B.5 MPEG-21 Multimedia Framework

The MPEG-21 vision is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by the different communities. The MPEG-21 multimedia framework will identify and define the key elements needed to support the multimedia value and delivery chain, the relationships between and the operations supported by them. MPEG-21 will elaborate the elements by defining the syntax and semantics of their characteristics, such as interfaces to the elements. MPEG-21 will also address the necessary framework functionality, such as the protocols associated with the interfaces, and mechanisms to provide a repository, composition, conformance, etc.

Seven architectural elements are identified as key to the multimedia framework. In addition, the user requirements within a multimedia framework are described separately as they impact upon each of the seven architectural elements. In summary the elements comprise:

- Digital Item Declaration: a uniform and flexible abstraction and interoperable schema for declaring Digital Items;
- Digital Item Identification and Description: a framework for identification and description of any entity regardless of its nature, type or granularity;
- Content Handling and Usage: provide interfaces and protocols that enable creation, manipulation, search, access, storage, delivery, and (re)use of content across the content distribution and consumption value chain;
- Intellectual Property Management and Protection: the means to enable content to be persistently and reliably managed and protected across a wide range of networks and devices;
- Terminals and Networks: the ability to provide interoperable and transparent access to content across networks and terminals;
- Content Representation: how the media resources are represented;
- Event Reporting: the metrics and interfaces that enable users to understand precisely the performance of all reportable events within the framework.

This report borrowed the concept of “feature” from MPEG-21 and MPEG-7 specifications and accommodated this level of granularity in the AIP structural map. In addition, the momentum of the MPEG-21 and MPEG-7 standards efforts was a contributing factor in the decision to design Extension Schemas as “plug-ins” to the Primary Schema thus allowing schema developed under MPEG-21 or MPEG-7 to be included in the AIP.

Appendix C Related Links

Appendix C contains the brief descriptions and URLs for websites used as resources for this report. The validity of the URLs was vetted at publication time, however, hotlinks from a Microsoft Word document do not always work as advertised. The most reliable approach is to cut and paste from this document to your browser.

Bristol University, An XML Encoding of Simple Dublin Core Metadata	http://www.ilrt.bristol.ac.uk/people/cmdjb/research/metadata/dc/xml/wd-dc-xml.html
CEDARS Project Home Page	http://www.leeds.ac.uk/cedars/
CORDIS IST Projects, IST-1999-11628 Broadcast Restoration of Archives through Video Analysis	http://www.cordis.lu/ist/projects/99-11628.htm
Corporation for Research and Education Networking Home Page	http://www.cren.net/
Council on Library and Information Resources Home Page	http://www.clir.org/
Digital Video Storage Chart	http://www.videotexsystems.com/videorate.htm
European Research Consortium on Informatics and Mathematics	http://www.ercim.org/
Joint Information Systems Committee, A Digital-Preservation Announcement and Information List	http://www.jiscmail.ac.uk/lists/digital-preservation.html
Library of Congress, AV Prototype Home Page	http://lcweb.loc.gov/rr/mopic/avprot/avprhome.html
Motion Picture Experts Group Home Page	http://www.cselt.it/mpeg/
National Archives and Records Administration, ARCHIVES II Project Home Page	http://www.sdsc.edu/NARA/
National Archives and Records Administration, Motion Picture Films and Sound and Video Home Page	http://www.nara.gov/research/bymedia/mo_int.html
National Gallery of the Spoken Word Home Page	http://www.ngsw.org/
National Information Standards Organization Home Page	http://www.niso.org/
National Library of Australia Home Page	http://www.nla.gov.au/
National Library of Australia, Digital Preservation Strategies	http://www-prod.nla.gov.au/padi/topics/18.html
National Library of Canada Home Page	http://www.nlc-bnc.ca/
Networked European Deposit Library Home Page	http://www.kb.nl/coop/nedlib/
Oxford University, Policy on Computer Archiving Services	http://info.ox.ac.uk/oucs/services/archiving/archive-policy.html
Oxford University, White Paper on	http://users.ox.ac.uk/~alex/hfs-AXIS-paper.html

Preserving the Electronic Assets of a University	
Reference Model for an Open Archival Information System ISO Archiving Standards Reference Model Papers including OAIS Reference Model	http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html
Research Library Group, Final Report and Recommendations on Preserving Digital Information	http://www.rlg.org/ArchTF/
Society of American Archivists Home Page	http://www.archivists.org/
Stanford University, Resources for Conservation Professionals	http://palimpsest.stanford.edu/
Universal Preservation Format (UPF) White Paper	http://www.rlg.org/preserv/diginews/diginews2-6.html#upf
University of Bath, Bibliography on Preservation of Electronic Information	http://www.ukoln.ac.uk/~lismd/preservation.html
University of California at Berkeley, Digital Library Home Page	http://sunsite.berkeley.edu/cgi-bin/welcome.pl
University of California at Berkeley, List of Image and Multimedia Database Resources	http://sunsite.berkeley.edu/Imaging/Databases/
University of California at Berkeley, Making of America II Home Page	http://sunsite.berkeley.edu/MOA2/
University of Illinois, An investigation of HDF as an archive format	http://hdf.ncsa.uiuc.edu/archive/hdfasarchivefmt.htm
University of Leeds, D-Lib Magazine, Article on Emulation as a Digital Preservation Strategy	http://www.dlib.org/dlib/october00/granger/10granger.html
University of Missouri, Digital Preservation Needs and Requirements in RLG Member Institutions	http://cecssrv1.cecs.missouri.edu/NSFWorkshop/digpres.html

Appendix D Glossary

Access Aids	Catalogs, finding aids, or indexes that allow Consumers to discover and retrieve Archival Information Packages of interest.
Access Method	A method for retrieving an Archival Information Package based on its name or identifier that is available to authorized users.
Access Management Extension Schema	An XML Extension Schema containing information about the rights and restrictions of access to an AIP. In an access management system, this information and information about the requesting Consumer is used to determine the extent and type of access permitted.
Access Subsystem	The OAIS functional entity that contains the services and functions that make the archival information holdings and related services visible to Consumers.
Administration Subsystem	The OAIS functional entity that contains the services and functions needed to control the operation of the other OAIS functional entities on a day-to-day basis.
Administrative Metadata	Information needed to support an organization's program administration and to support repository system administration and digital object management, including content preservation.
Archival Information Class (AIC)	A group of Archival Information Packages (AIPs) with common characteristics. In order to support the documentation or management of packages, information about the class or tools that work on the class are stored in a separate AIP whose identity is referenced in the metadata contained in each member of the class.
Archival Information Package (AIP)	From a physical perspective, the AIP consists of three components: metadata, data, and packaging. Each component consists of one or more files. The metadata component consists of XML Schema containing information describing the archival object. The data component consists of all the data files (essence bitstream s) that comprise the archival object. The packaging component encapsulates the metadata and data components, creating a single entity or "self-extracting archive" that is the AIP. An AIP contains both data files (essence bitstream s) and metadata. Once transmitted, an AIP may be de-constructed or otherwise treated to meet the needs of data management and the user community.
Archival Item	The original content item that is to be reformatted for management in a digital repository. In this report, the term is used synonymously with original

	source item. This document is strongly oriented toward a library program that reformats physical originals like phonograph records, videotape recordings, or digital compact disks into intangible-digital form. Original content, however, may be acquired in intangible-digital form and still require shaping into an AIP.
Archival Object	An abstract concept for a single unit of digital content, which may have been acquired by a library in digital form or which may consist of the digital representation or reproduction of a physical original. In the OAIS model, this content (the archival object) takes the form of a package, i.e., a submission information package, an archival information package, or a dissemination information package.
Archival Storage Subsystem	The OAIS functional entity that contains the services and functions used for the storage and retrieval of AIPs.
Archive System	A computer system designed to ingest, manage and store, and provide access to digital content. In this report, a repository is consider to be a system that will conform to the OAIS reference model
Audio Extension Schema	An XML Extension Schema containing technical metadata about the audio files in an AIP. In the Library of Congress project, the same XML schema may be used to contain technical metadata about multiple generations of audio files.
Class	A group or set of AIPs sharing common attributes (see Archival Information Class).
Collection	Archival items in a library that may be grouped because of their provenance (e.g., the George Washington Papers) or by intellectual properties (e.g., the folk music collections). In some cases, librarians may use the term to refer to a class of items, unified by some intellectual category. This latter usage is similar to usage in computer science, where a collection in an object-oriented system is a class that shares certain properties, although not necessarily intellectual characteristics.
Component	In the context of this report the term component is used to designate the parts of an AIP, which are data, metadata, and packaging.
Consumer	Persons or client systems that interact with a repository to discover, examine, and use content of interest.
Content Information	OAIS term for the content core of an AIP: a digital object and its Representation Information, e.g., the data files containing digitized audio and the corollary metadata describing the formats of these

	data files. Content information is the principal target for long-term preservation.
Context Information	OAIS term for the metadata that describes the relationship of Content Information to its environment, e.g., statements about why the Content Information was created and how it relates to other Content Information objects existing elsewhere.
Data	Information in its purest sense: a sequence of bits that may represent a table of numbers, the characters on a page, the sounds made by a person speaking, or the images comprising a motion picture. Used in contrast to metadata, information about information. Some broadcasters and entertainment media engineers use the terms essence or essence bitstream to name data.
Data Dictionary	A set of definitions for metadata elements, attributes, or database fields.
Data Dissemination Session	OAIS term for a telecommunications session that provides data to a Consumer. The format for the content delivered is negotiated between the repository and the Consumer in a Request Agreement.
Data Files	Bitstream s within an AIP that contain data as defined above. Strictly speaking, if an AIP is physically encapsulated, bitstreams do not take the form of files, but this report uses the term as if they were files.
Data Management Subsystem	The OAIS functional entity that contains the services and functions for searching, querying, and managing the information kept in Archival Storage.
Data Management Data	Data created and stored in the Data Management Subsystem.
Data Submission Session	OAIS term for a telecommunications session that submits content from a Producer for ingestion. The format for the content submitted is negotiated between the repository and the Producer in a Submission Agreement.
Descriptive AIC	The Archival Information Class (see definition above) whose members share descriptive information, or certain segments of descriptive information.
Descriptive Information	See Descriptive Metadata
Descriptive Metadata	Librarian's term for information that primarily describes content in intellectual terms and principally exists to support content discovery, sometimes synonymous with bibliographic information. Digital library specialists often use the term as synonymous with intellectual metadata.
Digital Audio-Visual Repository System	A computer system designed to ingest, manage and store, and provide access to audio and video digital

	content. In this report, a repository or repository system is consider to be a system that will conform to the OAIS reference model.
Digital Preservation	Sustaining the accessibility and readability of digital content for the long term, also stated as the preservation of digital content, and not to be confused with reformatting physical content into digital form.
Digital Repository	A computer system that ingests, stores and manages, and provides access to digital content. In the OAIS model, the four subsystems implied by the verbs in the preceding statement are joined by a fifth: administration.
Discovery	The process of finding desired information, including the use of search systems, indexes, and other access tools.
Dissemination Information Package (DIP)	The information package delivered to an authorized Consumer by an Access Module, consisting of elements drawn from an Archival Information Package and reshaped to fit this Consumer's requirements.
Encapsulate	To enclose as if in a capsule. In computer science, one example of encapsulation consists of packaging digital elements in self-extracting files, e.g., ZIP or TAR.
Environment AIC	The Archival Information Class (see definition above) pertaining to the computing environment in which members were produced or stored, and/or rematerialized.
Environment Schema	An XML Extension Schema containing metadata about the computing environment in which a class AIP were produced or stored, or may be rematerialized.
Essence Bitstream	Term used by broadcasters and entertainment media engineers to name instances of data, e.g., the files that contain representations of sounds, images, and the like.
Extension Schema	A separate namespace of the Primary Schema containing metadata too detailed or specific to be contained in the body of the Primary Schema.
External References	A pointer or reference to a location or address outside of the repository where additional metadata exists about an AIP.
Finding Aid	A type of access aid that supports user discovery processes. The term is used by librarians to name specialized forms of content description, often for large blocks of material, which may take a different form from conventional bibliographic or catalog records.
Fixity Information	Information that documents the authentication

	mechanisms and provides authentication keys to ensure that the Content Information has not been altered in an undocumented manner.
Format	For digital content, the term refers to the structure of an object, sometimes the structure of a particular bitstream or file, e.g., WAVE format audio, and sometimes for larger entities like AIPs, which will have a physical and a logical format.
ILS	Integrated Library System, e.g., the Voyager System in use at the Library of Congress.
Image Schema	An XML Extension Schema containing technical metadata about the image files in an AIP, or image files that may have been the source for the image files in an AIP.
Information Package	A content package transmitted between subsystems in the OAIS reference model. The generic of SIP, AIP, or DIP.
Infrastructure Independence.	The property of a system, subsystem, component, or data to function in a variety of computing environments.
Ingestion Subsystem	The OAIS functional entity that contains the services and functions that accept Submission Information Packages from Producers, prepare AIPs for storage, and ensure that AIPs become established within the archive system.
Inherit	In object-oriented programming, all members of a class inherit properties from a class description, and child objects inherit properties from parent objects in a hierarchy. In this report, the concept is applied to Archival Information Classes (represented by class-expressing AIPs), the properties of which are inherited by all of the AIPs that make up that class, and each member of a class will contain metadata pointing to the AIP that holds information about the class.
Intellectual Metadata	Librarian's term for information used to describe content that principally exists to support content discovery. Digital library writers often use the term as synonymous with descriptive or bibliographic information.
Intermediate Node	In this document, AIPs are viewed as hierarchical, and are made of Primary, Intermediate, and Terminal nodes.
LC	Acronym for the Library of Congress.
Logical View	A perspective or way of looking at the structure of an AIP that focuses on the relationships within the metadata component of an AIP.
Long Term	A period of time sufficient for there to be concern about the impact of changing technologies on a digital repository and thus affecting the content it

	contains.
MARC	MAchine Readable Cataloging format
MAVIS	Merged Audio-Visual Information System, a software product of Wizard Information Services, Ltd., used by the Motion Picture, Broadcasting, and Recorded Sound Division of the Library of Congress for the capture and management of metadata for selected categories of audio-visual material.
Metadata	Data that describes other data.
METS	Metadata Encoding and Transmission Standard, a successor to the Making of America 2 DTD, and associated with the Digital Library Federation and the Library of Congress.
Migration, format	The process by which the underlying bitstream s for a given unit of content are transformed from an obsolescent format into a new, successor format.
Migration, media or system	The process by which the underlying bitstream s for a given unit of content are moved from one device or storage medium to another, or from one system to another, without change.
Modular	The property of completeness and self-contained functionality in a system, subsystem, component, or data.
MPEG-21 **MPEG**	A recently emerging standard for audio-visual metadata from the Motion Picture Experts Group. The Motion Picture Experts Group, source of a number of standards, including MPEG-2 (a standard for encoding digital video in compressed form), MPEG-7 (a format for descriptive metadata), and MPEG-21 (an XML schema to serve as a framework for various other data sets, including MPEG-7). MPEG website: http://www.cselt.it/mpeg/
Namespace	A term used in XML to denote a section of an XML document wherein the naming of elements and attributes is self-contained; i.e. not global.
Navigate	The process by which a user moves logically or physically within information.
NISO	National Information Standards Organization.

OAIS	The Open Archival Information System reference model developed under the auspices of the National Air and Space Administration. The reference model is for a digital content archive or repository, in the words of the model, "consisting of an organization of people and systems that has accepted the responsibility to preserve information and make it available for a Designated Community." OAIS website: http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html
Original Source Item	The original content item that is to be reformatted for management in a digital repository. In this report, the term is used synonymously with archival item. This document is strongly oriented toward a library program that reformats physical originals like phonograph records, videotape recordings, or digital compact disks into intangible-digital form. Original content, however, may be acquired in intangible-digital form and still require shaping into an AIP.
Packaging	The process of binding and identifying the components of an AIP.
Packaging Information	That information that is used to bind and identify the components of an AIP.
Physical View	A perspective or way of looking at the structure of an AIP that focuses on the packaging of the AIP as a whole.
Preservation	Preservation in this document has two meanings. First, the term may be used in association with what library and archive professionals call reformatting, the making of "preservation copies" of endangered or deteriorating originals, for example by microfilming. Second, the term may refer to the persistence of digital content.
Preservation Description Information (PDI)	OAIS term for the information necessary for adequate preservation of the Content Information and which is categorized as Provenance, Reference, Fixity, and Context information.
Preservation Service Provider	An individual or organization responsible for operating and managing a digital repository.
Primary Node	In this document, AIPs are viewed as hierarchical, and are made of Primary, Intermediate, and Terminal nodes.
Primary Schema	The base set of metadata in the form of an XML Schema that describes the data contained within an AIP. In the context of this report, the Primary Schema is associated with the efforts of the Digital Library Federation to develop the Metadata Encoding and Transmission Standard (METS).
Producer	The role played by those persons or client systems

	that process born-digital content or reformat tangible content (analog or digital) into intangible digital form, ready for ingestion into a digital repository.
Production	The process of converting original source items into digital formats suitable for ingestion into a repository.
Prototyping Project	A multi-year effort to study aspects of a digital repository and associated aspects of the digital reformatting and preservation of audio-visual collections at the Library of Congress.
Prototyping Project Metadata Database	A Microsoft Access database currently used by the Prototyping Project to capture and manage digital-object metadata during production.
Provenance Information	Information that documents the history of an AIP, including the origin or source of the content, any changes made to the AIP, and the identification of the content managers.
Provenance Schema	An XML Extension Schema containing Provenance Information about an AIP.
Reference Information	Information that identifies, and if necessary describes, one or more mechanisms used to provide assigned identifiers for the Content Information. It also provides identifiers that allow outside systems to refer, unambiguously, to a particular Content Information.
Reference Model	A framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist.
Refreshment	A form of migration, media, or system (see above) in which unchanged bitstream s are refreshed by moving them to other media or devices.
Rematerialize	The process of presentation, display, or playback of an archival object which, after the passage of time, may require the emulation of obsolete content-presentation systems.
Repackaging	A form of migration, format (see above) in which alterations to bitstreams require changes to Packaging Information, or in which the Packaging Information itself has been changed.
Replication	A form of migration, media or system (see above) in which unchanged bitstream s and also the metadata are refreshed by moving them to other media or devices.
Repository	A multi-module computer system designed to

	ingest, manage and store, and provide access to digital content. In this report, a repository is considered to be a system that will conform to the OAIS reference model.
Representation AIC	The Archival Information Class (see definition above) whose members share Representation Information.
Representation Information	Information (or a pointer to information within metadata) that explains the structure and meaning of a given bitstream , e.g., ISO/IEC 4873:1991 Information technology; ISO 8-bit code for information interchange; structure and rules for implementation
Schema	A normative form of XML that describes the format and content of other XML.
Self-describing	A property of a system, subsystem, component, or module that denotes its ability to describe itself without external sources of information.
Self-extracting Archive	A form of packaging that provides the capability to encapsulate a set of files as a single file and subsequently allows for the extraction of one or more of those files for access, e.g. AIP or TAR.
SMIL	Synchronized Multimedia Integration Language (SMIL) is a layout language that allows creation of multimedia presentations consisting of multiple elements of compelling music, voice, images, text, video, and graphics in a common, synchronized timeline. SMIL was developed by a group from the CD-ROM, interactive television, Web, and audio and video streaming industries.
Structural Metadata	Information that supports binding and/or navigating a multi-part, complex digital object. In the proposed AIP, this function is executed by the Structural Map (part of the METS schema). Note that this definition differs from the usage in some engineering settings that refers to the inherent structure of a given bitstream.
Submission Agreement	OAIS term for an agreement reached between a Preservation Service Provider and a Producer that specifies rules for the Data Submission Session. This data model identifies format/contents and the logical constructs used by the Producer and how they are represented on each media delivery or in a telecommunication session.
Submission Information Package (SIP)	OAIS term for the Information Package identified by the Producer in the Submission Agreement with the Preservation Service Provider; compare to AIP and Dissemination Information Package.
Sub-object	A digital object within an AIP containing either

	metadata or data.
Terminal Node	In this document, AIPs are viewed as hierarchical, and are made of Primary, Intermediate, and Terminal nodes.
Text Schema	An XML Extension Schema containing technical metadata about the text files in an AIP.
Transformation	A form of migration, format (see above) in which there is an alteration to the Content Information or Preservation Description Information of an AIP, e.g., changing ASCII codes to UNICODE in a text document.
Video Schema	An XML Extension Schema containing technical metadata about the video files in an AIP.
XML	eXtensible Markup Language is a subset or dialect of Standard Generalized Markup Language. Its goal is to enable generic SGML to be served, received, and processed on the Web. The XML 1.0 specification was published as a W3C Recommendation on 10 February 1998.
XSL	Extensible Style Language is an XML-based language for transforming and reformatting one XML document into another XML document.