

The Making of America II Testbed Project White Paper

Version 2.0 (September 15, 1998)

READERS' GUIDE TO THE MOA2 WHITE PAPER.....	1
EXECUTIVE SUMMARY.....	1
PART I: PROJECT BACKGROUND.....	3
THE MOA II TESTBED PROJECT PLANNING PHASE.....	4
THE MOA II TESTBED PROJECT PRODUCTION PHASE.....	4
THE MOA II TESTBED PROJECT DISSEMINATION PHASE.....	5
PART II: THE MOA II DIGITAL LIBRARY SERVICE MODEL.....	5
AN OVERVIEW OF THE MODEL.....	5
A MODEL FOR DIGITAL LIBRARY OBJECTS	7
<i>Adding Classes and Content to the MoA II Object Model.....</i>	<i>7</i>
<i>Adding Metadata to the MoA II Object Model.....</i>	<i>7</i>
<i>Adding Methods to the MoA II Object Model.....</i>	<i>8</i>
Object Oriented Design (OOD) as Part of the Object Model.....	8
Defining the Difference between Behaviors and Methods.....	9
Methods as part of the MoA II Digital Object Model.....	10
BUILDING MOA II ARCHIVAL OBJECTS	10
SUMMARY	10
PART III: IMPLEMENTING THE SERVICE MODEL FOR MOA II.....	11
SELECTION OF MOA II DIGITAL ARCHIVAL CLASSES.....	11
THE MOA II TESTBED -- SERVICES AND TOOLS.....	13
BEHAVIORS AND METHODS—"WHAT TOOLS DO".....	14
<i>Definition.....</i>	<i>14</i>
<i>Contexts and Constraints.....</i>	<i>15</i>
<i>Navigation.....</i>	<i>16</i>
General Navigation.....	16
Image Navigation.....	17
<i>Display and Print.....</i>	<i>17</i>
<i>Combination or Comparison.....</i>	<i>18</i>
<i>Repository Search.....</i>	<i>19</i>
<i>Color Analysis.....</i>	<i>19</i>
<i>Bookmarks, Annotation, and Links.....</i>	<i>20</i>
MOA II METADATA	20
1) <i>Descriptive Metadata.....</i>	<i>20</i>
2) <i>Structural Metadata.....</i>	<i>20</i>
Structural Metadata Elements and Features Tables.....	21

3) <i>Administrative Metadata</i>	25
Administrative Metadata Elements and Features Tables.....	25
ENCODING – BEST PRACTICES.....	31
<i>Encoding Archival Object Content and Finding Aids</i>	31
<i>Encoding to Encapsulate Metadata and Content inside the Archival Object</i>	32
PART IV--BEST PRACTICES FOR IMAGE CAPTURE.....	33
SCANNING.....	33
DIGITAL MASTERS AND THEIR DERIVATIVES	34
IMAGE QUALITY	34
FORMATS.....	36
IMAGE METADATA.....	36
SUMMARY OF GENERAL RECOMMENDATIONS	36
SPECIFIC MINIMUM RECOMMENDATIONS FOR THIS PROJECT	37
APPENDICES.....	38
<i>Structural Metadata Notes</i>	38
BIBLIOGRAPHY	40
<i>Organization of Information for Digital Objects</i>	40
<i>Metadata</i>	40
<i>Scanning And Image Capture</i>	41

Readers' Guide to the MOA2 White Paper

The MOA2 White Paper is a complex document, which covers a broad range of topics, from overall guidelines in the collection of structural and administrative metadata and suggested best practice in imaging, to a detailed discussion of an object oriented approach in digital library construction and the development of tools to assist scholars. It is our hope that users will approach this paper in modules, and that those who wish to skip particular sections will be able to find other sections of use. This section was designed for readers who are interested in skimming portions of the MOA2 White Paper, as well those who wish to focus on detailed information on a particular topic.

Those who are interested in reading about the MOA2 Project and its goals should read the **Executive Summary**, and **Part I: Project Background**.

Those who are interested in the technical details of the Model for Digital Library Objects should read **Part II: The MoA II Digital Library Service Model**. Those who wish just an overview of the concept of the Model should read **An Overview of the Model** and the **Summary** from Part II. Those who are interested in a discussion of the use of Tools within the Digital Library should read **Part III: Implementing the Service Model for MoA II**. For a brief overview of this subject, see **The MoA II Testbed -- Services and Tools** within Part III.

Discussions of metadata and recommendations for collection of such are primarily discussed in Part III. Readers who wish to learn about structural metadata (i.e. metadata that are relevant to presentation of the digital object in terms of navigation and use) can consult the **Structural Metadata** section of Part III. Those who wish to follow the discussion of Administrative metadata (which we have defined as information used in the management of digital objects and collections) should see the **Administrative Metadata** section of Part III. This paper does not contain a detailed discussion of descriptive metadata.

Imaging best practices are discussed in **Part IV--Best Practices for Image Capture**. An overview of this topic along with specific recommendations can be found at the end of Part IV in **Summary of General Recommendations and Specific Minimum Recommendations for this Project**.

Executive Summary

The Making of America (MoA II) Testbed Project is a Digital Library Federation (DLF) coordinated, multi-phase endeavor that proposes to investigate important issues in the creation of an integrated, but distributed, digital library of archival materials (i.e., digitized surrogates of primary source materials found in archives and special collections). This paper is a milestone in the MoA II planning phase and identifies a starting point for the testbed that will be created in the production phase of this project, with funding from the National Endowment for Humanities. An overview of this paper's goals and the MoA II project is contained in this executive summary. Detailed project background information follows in the next section of this paper.

The library community has a distinguished history of developing standards to enhance the discovery and sharing of print materials (e.g., MARC, Z39.50, ISO ILL protocols, etc.). This leadership role continues today through library participation in creating new best practices and standards that address digital collections and content issues (e.g., EAD, TEI, preservation imaging, etc.). In addition, libraries have worked actively within the broader Internet community to adopt other standards that are used to store and access digital library materials (e.g., TIFF, HTTP, URNs, etc.). Perhaps the most important goal of this paper is to open a new dialogue in the ongoing conversation about digital library standards, specifically, to discuss the need for any new best practices and standards that are required if the digital library is to meet traditional collection, preservation, and access objectives.

The discussion this paper hopes to stimulate builds on work completed to date and asks the question, “How can we create digital library services that interoperate in an integrated manner across multiple, distributed repositories?” Clearly, the standards and best practices mentioned above play an important role in answering this question. However, this paper and the MoA II Testbed Project in general focus on a new area of discussion that goes beyond the discovery of a digital object, and focuses on how it is handled once it is found. That is, the paper and testbed focus on the need to develop standards for creating and encoding digital representations of archival objects (e.g., a digitized photograph, a digital representation of a book or diary, etc.). If tools are to be developed that can work with digitized archival objects across distributed repositories, these objects will require some form of standardization.

This paper aims to begin the discussion of digital object definitions by developing and examining metadata standards for digital representations of a variety of archival objects, whether they be in the form of text, digitized page images, photographs, etc. For our purposes there are three types of metadata: *Descriptive*, *Structural*, and *Administrative*. *Descriptive metadata* is used to discover the object. The project testbed proposes to use existing descriptive metadata standards (such as MARC records and the Dublin Core), as well as existing descriptive/structural metadata (like the EAD) to help the user locate a particular digital object. The paper proposes defining new standards for the *Structural* and *Administrative* metadata that will be needed to view and manage digital objects. *Structural metadata* defines the object’s internal organization and is needed for display and navigation of that object. *Administrative metadata* contains the management information needed to keep the object over time and identify artifacts that might have been introduced during its production and management (e.g., when was the object digitized, at what resolution, who can access it, etc.).

At a higher level, this paper proposes a Digital Library Service Model in which services are based on tools that work with the digital objects from distributed repositories. This borrows from the popular object oriented design model. It defines a digital object as encapsulating content, metadata and methods. *Methods* are program code segments that allow the object to perform services for tools, such as “get the next page of this digital diary.” Unlike other models, methods are included as part of the object. This paper proceeds by identifying several archival digital object *classes* that will be examined as part of the MoA II project, including photographs, photo-albums, diaries, journals, letterpress books, ledgers and correspondence. One of the first development efforts for the testbed will be to create the tools that display and navigate these MoA II objects, some of which have complex internal organization. Therefore, another goal of this paper is to identify the structural metadata elements that are needed to support display and navigation, to ensure they are included as part of the digital objects. In addition, this paper begins to examine the methods (program code) that could be included with each class of object.

Because each partner library in the MoA II project will digitize images, the paper also investigates issues around best practices for digitization, in particular the capture of administrative metadata as part of this process.

After this paper has been reviewed by the wider community, the MoA II participants plan to incorporate reader feedback into the development of digital object definitions for the classes of materials to be examined in the MoA II Testbed. These definitions will specify how to encode the content, metadata and methods as part of the object. An important goal of the project is to use the testbed to investigate the advantages and limitations of these definitions and help stimulate a broader discussion of standards for digital library objects and best practices for digitizing archival materials. This discussion must include the project participants, the DLF membership and the wider community. In addition, the project will contribute to the existing discussion in the DLF Architecture Committee on distributed system architectures for digital libraries. The MoA II testbed will give the library and archival community a tool that can be used to test, evaluate and refine digital library object definitions and digitization practices. It is expected that these discussions will move the archival community and the library community in general, closer to consensus on standards and best practices in these areas.

Part I: Project Background

The Digital Library Federation has coordinated a grant proposal that requests support to develop a testbed for its Making of America II Project. This proposal was submitted to the National Endowment for the Humanities (NEH) by UC Berkeley and includes the participation of five DLF members: Cornell University, New York Public Library, Pennsylvania State University, Stanford University, and UC Berkeley. The objective of the MoA II Testbed project is to move the DLF membership, as well as the wider community, closer to the realization of a national digital library by addressing a number of issues that are on the critical path to this goal. The MoA II Web site can be found at <http://sunsite.Berkeley.EDU/moa2/>.

Specifically, the MoA II Testbed will provide a vehicle that will allow the DLF to investigate, refine and recommend metadata elements and encodings used to discover, display and navigate these digital archival objects. In addition, this project will provide a guide to best practices for the digitization of archival materials. Overall, the DLF expects that the MoA II testbed will provide a working system in which metadata and digitization problems can be investigated, and where different solutions can be discussed, tested, and refined. The project will provide the DLF membership with information that could be used to create standards or best practice recommendations for each research area, all of which are required for the creation of a national digital library. In addition, the project will be of great value to the larger library community, in that it will advance the discussion of the nature of the digital library and move the community toward consensus as part of our ongoing discussion.

This project is broken down into three phases. The project planning phase has been funded by the DLF and is now underway. The NEH-funded production phase will commence in July. This is where we will be able to test the theories developed in the planning phase. Finally, at the completion of the production phase, the project will disseminate its tested ideas and practices to the broader community.

The MoA II Testbed Project Planning Phase

The MoA II Testbed proposal, submitted to the NEH for funding in May 1998, included a planning phase to be funded by the DLF that covers the time directly preceding NEH funding. UC Berkeley has now received funds from the DLF that will allow the planning phase activities to proceed. The MoA II planning year proposal to the DLF can be found at <http://sunsite.Berkeley.EDU/moa2/moaplan.html>.

It is crucial that the methodology employed by the MoA II Testbed Project engage the wider community of scholars, archivists and librarians interested in access to the digital materials represented in this project. In addition, this process must also include metadata and technical experts at the proper time to ensure their contributions are utilized to maximum effect. Therefore, the following methodology is recommended. These activities include:

1) UC Berkeley, working directly with the other four NEH participants, consultants and selected archivists, will review the collections that have been proposed for conversion and identify the classes of digital archival objects to be represented in the testbed. The classes could include formats such as correspondence, photographs, diaries, ledgers, etc. Note: The MoA II Steering Committee has recommended that books and serial articles be considered out of scope for this project.

2) UC Berkeley, working directly with the other four NEH participants, consultants and selected archivists, will draft a white-paper that identifies the *behaviors* each class of digital objects should be able to exhibit, as well as the structural and administrative metadata to support these behaviors. In addition, the white paper will suggest initial best practices for digitizing the classes of archival objects to be included in this project. The white paper will include a compilation of existing work in all the above areas, as well as any original contributions the group can provide.

3) The participants of the MoA II Testbed project and the DLF Architecture Committee will review the white paper. Upon revision from comments received, the paper will then be available for distribution as a basis for discussion in the wider community.

4) Technical experts on the Berkeley staff will analyze the white paper and design a means of encoding the behaviors, metadata, and objects for implementation during the Research and Production Phase of the project.

The MoA II Testbed Project Production Phase

The MoA II testbed will be created in the NEH-funded year and will be used to investigate, refine and enhance the working definitions of administrative and structural metadata, the key behaviors of archival objects and best practice guidelines for digitization. The goals of the testbed are to:

- Create tools that help the community understand how digital archival objects are discovered, displayed and navigated;
- Understand how metadata is used by these tools and come to a better understanding of what value the metadata provides, and at what cost and;

- Provide the DLF a set of metadata practices that can be reviewed and recommended to the wider community.

The MoA II Testbed Project proposal submitted to the NEH can be found at <http://sunsite.Berkeley.EDU/moa2/moaproposal.html>.

The MoA II Testbed Project Dissemination Phase

Following completion of the Research and Production Phase, the MoA II project will seek funding for an invitational seminar to review project results. Participants will include representatives of a broad spectrum of fields and interest groups, including, for example, digital library experts, archivists and special collections librarians, scholars, computer scientists, museum technologists, and others who have participated in other phases of development of the EAD protocols, are engaged in similar work, or who have appropriate expertise. The results of this phase will include widespread dissemination of the results of the project, refinement as necessary of the practices established, and formulation of an agenda for further community review and acceptance.

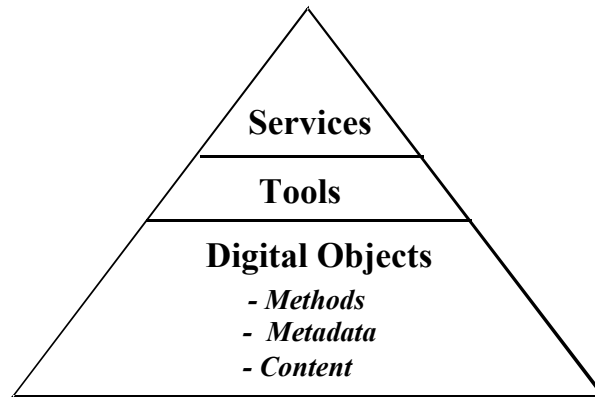
Part II: The MoA II Digital Library Service Model

An Overview of the Model

The *Digital Library Service Model* developed for the MoA II Testbed Project has three layers (Figure 1), one each for *services*, *tools* and *digital objects*. In this model, services are performed through tools that discover, display, navigate and manipulate digital objects from distributed repositories.

This paper also proposes a *Digital Object Model* that fits within the overall Service Model. The Object Model defines digital objects, which are the foundation of the Service Model, as an encapsulation of content, metadata and methods.

Figure 1: Digital Library Service Model



1) The Service Layer

This top layer describes the services that are to be provided for a specific audience of users. Given the MoA II project is focused on the use of archival materials by scholars, these services could include the discovery, display, navigation and manipulation of digital surrogates made from these collections. The specific service model used in this project follows the standard archival model. That is, materials can be discovered via USMARC collection level records in a catalog; the catalog records can link the user to the related finding aid that describes the collection in more detail and; the finding aids can link to individual digitized archival materials.

The service layer is actually comprised of a *suite of tools* that is created to support the needs of a particular audience. For example, scholars would be comfortable using sophisticated electronic finding aids to locate and view digital archival materials such as photographs or diaries. However, fifth-graders, with less rigorous information needs, may require simpler tools to discover and view these items.

2) The Tools Layer

This layer contains the tools that act at the request of the user. For example, a tool may be created to display and navigate a diary. The MoA II tools will consist of:

- an online catalog used for the discovery and display of the USMARC collection level records;
- a SGML compliant database used to search, display and navigate the Encoded Archival Description (EAD) compliant electronic finding aids and;
- various tools to display and navigate the MoA II compliant digital archival objects. Note: the objects are said to be MoA II compliant when they can be delivered using the proposed encoding standards described later in this paper.

Any tool itself is actually a *suite of behaviors*. That is, behaviors represent actions the tool can take on behalf of the user. In our digital diary example, behaviors could include actions such as turn to the next page, the previous page, jump to chapter three, translate this page to French, etc.

3) The Digital Object Layer

This layer contains the actual digital objects that populate distributed network repositories. Objects of the same *class* share encoding standards that encapsulate (i.e., includes) their content, metadata and methods – a full explanation of this concept follows in the next section. Separate digital object classes could be defined for books, continuous tone photographs, diaries, etc.

A Model for Digital Library Objects

Digital library objects form the foundation layer of the Digital Library Service Model, as described in the previous section. We can now create a Digital Object Model for these objects that will fit within the overall Service Model.

Adding Classes and Content to the MoA II Object Model

The MoA II object model defines *classes* of digital archival objects (e.g., diaries, journals, photographs, correspondence, etc.). As expected, each object in a given class has content that is a digital representation of a particular item. The format of the content can be digitized page images, ASCII text, numeric datasets, etc. The following examples describe three classes of archival objects, along with their content format.

- *Photograph* made up of a single digitized TIFF image;
- *Photo Album* made up of 30 photograph objects;
- *Diary* made up of 200 digitized TIFF page images and textual transcriptions.

The Object Model starts by defining classes of archival objects, where each object has content that is an electronic representation of a particular archival item of that class.

Adding Metadata to the MoA II Object Model

For the purposes of this discussion, we will consider metadata as separate from content. Metadata is data that in some manner describes the content. The DLF systems architecture committee has identified three type of metadata:

1) *Descriptive Metadata* is used in the discovery and identification of an object. Examples include MARC and Dublin Core records.

2) *Structural Metadata* is used to display and navigate a particular object for a user and includes the information on the internal organization of that object¹. For example, a given diary has three volumes: volume one is comprised of two sections called *dated entries* and *accounts* respectively; the entry section has two-hundred entries; entry twenty is dated August 4th, 1890 and starts on page fifty of volume one.

¹ Structural metadata could exist in various levels of complexity. The diary example above represents a rich structure that may be created for an important work and would include a transcription of the digitized, handwritten pages. The structure of the diary could be encoded in this transcription and the structural metadata could be extracted from the same. On the other extreme, a diary could exist with only enough structural metadata to turn the pages.

3) *Administrative Metadata* represents the management information for this object: the date it was created, its content file format (JPEG, JTIP, etc.), rights information, etc.

We can now add metadata to our model by stating that any given class of archival object encapsulates both the content and metadata, where the metadata is used to discover, display, navigate, manipulate and learn more about a particular object's management information.

One final note on metadata recognizes that the distinction between the types of metadata is not absolute. For example, chapters are part of the structure of a book, but chapter headings may be indexed to aid in the discovery of this item, thus filling one of the roles of descriptive metadata. In fact, the text of a book itself could be indexed and used for discovery.

Adding Methods to the MoA II Object Model

Methods are a concept defined within the object oriented analysis and design paradigm. Therefore, it would be useful to begin by reviewing concepts to be used in this paper that originate from object oriented design.

Object Oriented Design (OOD) as Part of the Object Model

Object oriented design has become very popular, as can be seen by the widespread use of related programming languages like C++ and Java. Some of the reasons for this popularity also make OOD an attractive addition to the Digital Library Service Model. In particular, object oriented design actually models users' behaviors, making it easier to more accurately translate their needs into system applications. This advantage will be discussed in more detail in the following section.

There is another important advantage for considering OOD. In object oriented design, a digital object conceptually *encapsulates* both content and methods. An object has content, as expected, but it also contains segments of program code called *methods*. These methods are part of the object and can be used by developers to interact with the content. For example, a developer can ask a digital book object named Book1 for page 25 by executing that object's *get_page()* method and specifying page 25. This method call may look something like *Book1.get_page(25)*.

There are a number of advantages in making methods part of the object, but probably the most important is that these basic program segments do not have to be re-invented by every developer creating a new tool². Instead, the developer can have the tool ask the object's existing method to perform the work needed. The ability for tool developers to "reuse" these methods makes the development of new tools faster and easier.

² *Technical Note:* It is worth noting that the above digital object model is only a conceptual model. In fact, complete objects made up of metadata, data and methods would not sit in a repository waiting for use. Instead, they are created as needed. That is, the parts of the objects (i.e., methods, metadata and content) are assembled from different areas of persistent store located anywhere on the network. Using the object oriented model does not require a repository to use specific object technologies like object oriented databases. Relational databases, for example, could be used for the persistent storage.

Since tools directly support the end user in this model, we want to encourage their development as much as possible!

Defining the Difference between Behaviors and Methods

One great advantage of the object oriented design approach is that it models users' behavior with methods. Therefore, this paper will now introduce a clear distinction between user level *behaviors* and *methods*. Simply put, behaviors are how users describe what tools can do for them. For example, zoom in on this area of a photograph, show me this diary, display the next page of this book, or translate this page to French. Methods are how system designers describe what tools can do for a user.

One important purpose for distinguishing differences between user level behaviors and methods is to put in place a process where the library community can engage their users in a dialogue on what services and tools they require, down to the behaviors they need in each tool. Software engineers can then map the user behaviors into sets of methods that are required to perform the necessary functions. The line between behaviors and methods represents the transition from user requirements to system design.

The following are example user level behaviors that might be relevant to a digital library object class of type diary.

- Show me the organization of this diary (e.g., it may have three volumes, each of which includes a section on dated entries, accounts and quotes)
- Show me the first page of Volume 1
- Show me page 3, the next page or the previous page
- Show me the fourth journal entry
- Show me the first entry for August 1890
- Show me more entries on the same topic as this one
- Show me entries that are separated by gaps of more than 10 days
- Show me entries that have these words in them
- Bookmark this entry
- Annotate this entry
- Share these entries with my colleagues

In each of the above cases, these users level behaviors would have to be mapped into a series of methods that performed the behavior. A short example may help illustrate the mapping that occurs between behaviors and methods. Imagine a user level behavior that is described as, "show me this diary." The tool executing this request could use object methods to: *a*) fetch the table of contents; and *b*) fetch the first page of the diary. The tool would then use its own methods to display the table in one browser frame and the first page in another frame.

Methods as part of the MoA II Digital Object Model

We can now add methods to the Object Model. At this point, it is important to note the close relationship between methods and metadata. In most cases, the methods require that appropriate metadata to be present if they are to perform their functions³.

The MoA II Object Model includes methods that are conceptually encapsulated along with content and metadata within an object of any given class, where the methods are used by tools to retrieve, store or manipulate that objects content. Methods often need the object's metadata to perform their functions.

Building MoA II Archival Objects

The final step in building a digital library object is to encapsulate the methods, metadata and content (i.e., data) into a digital library object⁴. The metadata and content must be *encoded* in a standard manner for objects in a given class. This is required so the methods (programs) that are defined for each class can work across all objects in that class.

Summary

This paper proposed a *Digital Library Service Model* for the MoA II project in which *services* are based on *tools* that work with the *digital objects* from distributed repositories. This model recommends that libraries first define the services they need to provide for each audience they support, then define the tools that are needed to implement these services. This process should include the identification of the tools' user level *behaviors* (i.e., what the tools do as required by the users).

This paper also proposes a *Digital Object Model* that fits within the overall Service Model. The Object Model describes digital objects, the foundation of the Service Model, as an encapsulation of *content*, *metadata* and *methods*. Different classes of objects exist (e.g., diary, photograph, etc.) and the content of each object can be text, digitized page images, photographs, etc. The object also contains metadata that is divided into three types: *Descriptive metadata* used to discover the object; *Structural metadata* that defines the object's internal organization and is needed for display and navigation of that object; and *Administrative metadata* that contains management information (e.g., the date the object digitized, at what resolution, who can access it, etc.). Finally the digital object definition borrows from the popular object oriented design model and includes methods as part of the object. *Methods* are program code segments that allow the object to perform services for tools, such as "get the next page of this digital diary."

³ The methods that are part of an object will tend to be the ones most used across sets of tools. Tools themselves will have methods and therefore, will need access to the metadata and content of the objects. We expect that every object will have a base set of methods that can provide the tools with any metadata or content that is required.

⁴ Technical note: While the content and metadata need to be encoded in a standard manner, they do not necessarily have to be stored together. In fact, the three different types of metadata do not need to reside together. This is due to the fact that objects only come into existence on an as needed basis. Therefore, the object can be "assembled virtually" from persistent storage when required.

Part III: Implementing the Service Model for MoA II

Selection of MoA II Digital Archival Classes

From the materials suggested by MoA II participating institutions, we have selected a group of object types/classes (limited only so that we will be able to complete the testbed within the timeframe of the project). We are proposing that these form the core of archival object types that will be examined in this project. The absence of something from the list does not mean that it cannot be digitized; it can be included as an object that will only be navigable in the simplest sense.

Continuous tone photographs: Single archival object. May have caption or other textual information recorded on its face or on verso. Continuous tone photographs are interesting for this project for a number of reasons: they exist in quantity in many of our collections; and they will give us an opportunity to look closely at the collection of administrative metadata for use in object behaviors. The most basic of objects, the continuous tone photograph will help us to build a solid platform upon which we can base the rest of our work.

Photo albums: Bound manuscript object, containing a collection of continuous tone photographs. The photo album may contain captions, which are separate from the photographs or other items, such as newspaper clippings, etc. Photo albums are in a way a logical extension of continuous tone photographs, since they contain photos ordered in a structured manner, allowing us to look at structural metadata issues, in addition to administrative metadata.

Diaries, journals, and letterpress books: Bound manuscript objects, usually arranged chronologically and with date notations. May have additional structure, such as an accounts section noted in the back. Again, these are structured documents, with the further possibility of additional metadata in the form of partial text (dates and other markers) included for additional navigation. With the inclusion of full texts, such as the William Henry Jackson diaries at NYPL, full searching and navigation is possible.

Ledgers: Bound manuscript objects that contain accounting records, usually arranged by account, although sometimes they are also in simple chronological order. Clearly, documents of this sort have a different structure than diaries and journals, but (from the structural/navigation standpoint) are they really a different object type, or a variation on a theme? Again, inclusion of more text, while more costly, allows for more sophisticated searching and navigation.

Correspondence: Objects of this class may be simple (a single page letter) or complex (a letter with an envelope and enclosures and/or attachments). Investigating correspondence will allow the project to examine these sometimes complicated documents and the structural metadata relationships between the sub-documents (letter to envelope, for example).

The classes of material listed above have been selected for the testbed either because large quantities are held by participating institutions, and/or because they offer the MoA II Project important challenges in terms of the behaviors needed to view, navigate, or manipulate them. The complex structure of photo albums challenges us to offer the viewer the ability to see individual photographs, to see a photograph with its caption, and to see photos and captions within the context of pages and pages within the context of an entire album.

The complex structure of diaries and journals allows us to explore presentation to the user of individual entries and let them jump from one entry to another (or from an index to an entry). It also lets us explore issues raised when individual page scans have no correspondence whatsoever with the logical structure of the document (i.e. journal and diary entries frequently end in the middle of a physical page and a new entry begins on that same page). In addition, these materials allow for display and navigation experiments when different levels of metadata are available. For example, a “minimal” digital diary might be comprised of a series of page images, in which case, only a base set of behaviors that can be implemented (e.g., turn to the next page, previous page). However, a richer diary may have encoded text transcribed for each page image that allowed for tool behaviors that can; display a table of contents for the diary; jump to a particular page or dated entry; search for text strings or “more entries like this one;” etc.

The structure of letterpress books and ledgers lets us examine the interaction between indices that exist within a document and the various individual entries/parts in that document. The project will also explore how the structure of these items differs from those of diaries and journals. While ledgers, letterpress books, journals and diaries clearly are different classes of items from an archivist’s point of view, they may be more similar than not from a structural metadata perspective.

In addition, the MoA II testbed will give the participants and the wider community a chance to evaluate different practices for encoding the relationships between objects. In particular, it will help the community understand the advantages and drawbacks of using these practices based on how tools are able to implement different behaviors for each practice. For example, a series of correspondence could be scanned and:

- a) placed into a single “base object”;
- b) created as separate objects, one for each letter, and then all linked together through the creation of a new “aggregate collection (folder?) object”;
- c) created as separate objects inside a “embedded collection (folder?) object.” A collection object has metadata for the collection, followed by the embedded objects, each with their own metadata and content. This differs from b) in that the objects are embedded, as opposed to linked;
- d) organized through a finding aid in which the container list points to any of the above.

Each base object, whether it stands by itself or is part of an aggregation or embedded collections object, can have “divisions.” That is, it can be divided into sections through the use of text encoding. For example, a diary can have dated entries that are identified by the text encoding which can then be used for display and navigation. In the same manner, any type of object can also be a “compound object” by linking to other objects, or embedding them inside itself. While the concepts of compound, linked or embedded objects is not new, the MoA II testbed will give the archival and library community a tool to better evaluate all the above options for digital archival objects, particularly in the context of distributed repositories.

The MoA II testbed will give the DLF and the wider community an opportunity to create objects using all the practices listed above. As important, it will allow for the evaluation of each practice, as they can be better understood based on how tools can use each practice to meet the needs of their intended audience.

The MoA II Testbed -- Services and Tools

The Digital Library Service Model described earlier in this paper proposed a three-tier model, with services, tools and digital objects comprising the layers from top to bottom. The MoA II testbed will implement the standard archival model within the Digital Library Service Model. That is, USMARC collection level records in a catalog will link to their related finding aids, which in turn will link to the digital archival objects in that collection.

The top tier in the model represents a service layer that is comprised of *suites of tools* that focus on supporting particular audiences. (e.g., scholars, university undergraduates, K-12 students, etc.). For example, archivists may require different tools for the discovery, display, navigation and manipulation of digital archival objects than would K-12 students. The MoA II Testbed project will initially focus on general services for scholars in using the classes of digital objects selected for this project. Future research projects that could be considered may include developing service models for more novice users (e.g., K-12) or more customized services for specialized scholarship (e.g., service for medieval manuscript scholars, as envisioned by the Digital Scriptorium Project).

The suite of tools to be developed in the MoA II testbed will initially include:

- An online catalog (OCLC's SiteSearch) used to discover and display the USMARC collection level records;
- A SGML compliant database (INSO's DynaWeb) used to search display and navigate the EAD compliant electronic finding aids and;
- Display and navigation tools to be used with MoA II compliant digitized photographs, photo albums, diaries, journals and correspondence. As the project enters the production year, we will solicit input from various concerned parties (archivists, scholars, librarians, etc.) to understand the behaviors required in this tool-set.

The MoA II testbed implementation of the Digital Library Service Model is now represented in Figure II.

Figure 2: MoA II Model Implementation



The goals of the testbed are to: create tools that help the community understand how digital archival objects are discovered, displayed and navigated; understand how metadata is used by these tools and to come to a better understanding of what value the metadata provides, and at what cost; and to provide the DLF a set of metadata practices that can be reviewed and recommended to the wider community.

Behaviors and Methods—“What Tools Do”

Definition

The Digital Library Service Model presented earlier defines **behaviors** as the way that “users describe what tools do for them.” Engaging users in a dialog about behaviors is a methodology that can be used to understand user needs as defined by the functionality of the tools. These user-level behaviors are then mapped by system designers into **methods**—discrete segments of program code that execute operations for tools. The translation from behaviors to methods represents the transition from defining user needs to system design. In many cases, high level methods in a tool will have the same name as a user level behavior. Creating methods that model a user’s desired behaviors is, in fact, one of the strengths of object-oriented design.

When we speak about “methods,” we speak about the operations that tools let us perform on digital objects. We will use this notion of methods to specify the range of activities that should be supported through the metadata described elsewhere in this white paper. In an object-oriented model, the methods are embedded in the digital objects; digital objects reveal methods to tools interacting with them. We acknowledge that many repositories in the MoA II environment will not deploy object-oriented models. Instead, the methods will be made available to tools that interact with *repositories* rather than the individual digital objects themselves. Nevertheless, we believe that this model is helpful

both in conceptualizing the nature of the tasks supported and in preparing for a type of digital library that may scale more effectively than current architectures.

Methods supported in the digital library should be both those common and exceptional operations users expect to perform with the digital objects. For an image collection, a method might be facilitating a “pan” or “zoom” on a portion of an image, or providing an enlargement of an image. For an encoded diary, the digital object’s method might involve providing the tool information about levels and types of organization (e.g., one volume, including 128 dated entries, an itinerary, and a list of contacts with addresses). The encoded diary’s methods might also yield both simple (“next entry,” “previous entry”) and more complex navigation (e.g., locate the first dated entry in November 1884; find entries where dates are separated by more than ten days). While no object or repository would be *required* to support the full range of methods—a practical impossibility—the model proposed here will facilitate the development of increasingly sophisticated tools that can scale for use on a growing body of complex archival objects.

Contexts and Constraints

The **methods** of the digital library reside in tools that are sometimes client-based and sometimes server-based, depending on the state of our technology and that of our users. The location of that method (i.e., on client or server) may shift with changes in those circumstances. For example, widespread adoption of an image client that supports progressive transmission of image data might shift image processing from server to client, expediting image processing and reducing load on the server. However, an interim measure might rely on a server-based compression/decompression process where the server can generate “pan” or “zoom” views at the user’s request in real time, thus relieving the client of processing responsibility and shifting the work to the server.

Just as the methods we discuss may move from client to server and back again, so too will they separate into specialized functions or merge into high level, multi-faceted functions. For example, we might postulate that “print” and “display” are different methods, with one object optimized for screen display, and another object optimized for a printer. While being careful to make clear that the authors of this document are *not* endorsing Acrobat as a tool or, especially, PDF as a storage format, we can see that Acrobat demonstrates a model where “display” and “print” are merged in the same tool. If we argue that Acrobat handles “display” poorly, we might push for a clearer separation of these two methods. By separating the methods conceptually, we are able to assess the applicability of the tool in the service of the method.

High level methods are frequently comprised of a series of calls to lower level methods. For example, because “print” is a behavior most users require in a tool, most tools will have a high level method expressed something like “print(a1,a2...aN).” In the example, the information inside the parentheses represents arguments that tell the method what object to print, which printer to use, etc. The “print” method would actually execute a series of lower level methods. It may, for example, first ask an object to deliver its content in a format suitable for printing by executing the proper method. Next, it may execute an operating system specific method that sends (i.e., spools) the formatted content to that particular printer.

Finally, some methods are applicable to all or most objects, while others may need to be finely tailored to the type of object—so finely tailored, in fact, that they rely on entirely different functions or primitives. An obvious example is the difference between

navigation of pages and navigation of portions of an image. A system that navigates a bound book might use operations such as “display next page” or “show page list”; a system that navigates a continuous tone image might use operations such as “display 200x200 pixels centered on coordinates X by Y”. The following sections attempt to define methods that we believe are central to creating the digital library. Wherever possible, we will attempt to describe methods in ways that are sufficiently generic to be applicable to a wide variety of objects; in some cases, we will describe methods specific to some data types.⁵

Navigation

General Navigation

We can think of **navigation** as a process stream consisting of a *request*, a *receive*, and a *display* action, where each action interacts with a **reference to objects** or metadata for objects rather than to the objects themselves. A significant portion of the user’s activity is what we would think of as navigation. For example, the user who finds a digital scrapbook in a repository will request what we would call a “table of contents.” Depending on the extent to which the book was processed, that table of contents may consist only of a stream of page references, or it may consist of a nested list of chapter and section headings. In navigating through the scrapbook, the user’s navigation tool will:

- 1) *request* references (to pages listed by page number or to sections listed by section headings);
- 2) *receive* the references in a discernable format; and
- 3) *display* that information in a meaningful way for me as the user of the tool.

Importantly, the user expects a series of **references**, and not actual delivery of the objects (i.e., the actual pages or sections will not be sent until requested).

Navigation also depends on an understanding of *relationship primitives*. These relationship primitives, *parent*, *child*, and *sibling*, are the generic references a tool uses to facilitate navigation. The navigation method is affected by the tool requesting, receiving, and displaying the parents, children, or siblings of an object that the user has located. For example, to navigate a fully encoded and logically organized scrapbook, a navigation tool might request references to the first level children in the scrapbook. A user would be presented with a list that looked like the following:

- Dated Entries
- Accounting/Budgetary data
- Names and Addresses
- Itinerary

Each of these major headings may be presented as links for further expansion; they would perhaps in turn offer second-level headings to the user (e.g., annual groupings of entries in the “dated entries” section). A threshold setting in the navigation tool may instruct the repository to send no more than N references at a time, and the repository would make a determination that, for example, all first level and second level headings fit within the threshold (i.e., thus providing the annual groupings within the “dated entries”

⁵ Clearly, “generic” and “specific” are difficult to define in this context. We hope that the discussions that take place in the MoA II process will help define methods we can agree are “generic.”

section at the same time it provides the four major headings listed above). At some point, the children of a given parent will be links to the objects themselves. We can imagine, then, in the example above, being presented with a navigation list of dated entries, any of which could be selected for **display**. Types of object references vary depending on the type of resources, the amount of funding available to process the materials, as well as programmatic or other purposes of an initiative. Examples include conceptual and structural references (as in the example above), simple page lists (cf. Project Open Book and the UM Making of America sites), and pages of thumbnail representations of larger format images.

Image Navigation

In contrast to the more generic form of navigation discussed above, image navigation uses image-specific information to accomplish its ends. Systems that display image information need information analogous to *geographic* references—X and Y coordinates along with dimensions of the portion of the image to be displayed. Increasingly, tools for image management and manipulation use notions of segmentation to optimize the relatively confined space of a video display, as well as other resources in short supply (e.g., network capacity, memory, and CPU capacity). As mentioned earlier, some of these technologies are primarily server-based, while others shift responsibility to the client. Wavelet compression, for example, allows a repository to store a form of the image rich in information (e.g., extremely high resolution), and to generate lower resolution versions and subsets in real time, at the request of the user or an intermediary. Another approach implicit in the tool/format called JTIP segments the image into overlapping tiles in a pyramidal structure, allowing the user to pan and zoom on a full resolution image by requesting the next tile, or a corresponding tile at a higher resolution. In both of these cases, the image **navigation** tool receives information about resolutions, resolution ratios, and window sizes to make the navigation possible; the image **display** tool (discussed below) uses that information to pan, zoom, crop, and otherwise use images.

Display and Print

The **display** method uses a reference to a known object to effect delivery of an item to a screen-oriented tool (e.g., a graphical web browser). By contrast to navigation where the user tool requests object references, the **display** tool will work from an object identifier it has received from an intermediary or that it can infer from a query where only one object exists. From this relatively simple operation, we quickly encounter more complex issues.

One of the most complex issues for **display** is the variety of known item references that an intermediary must handle. At its simplest, a display tool will encounter references to page images in a format clearly identified by structural metadata. Similarly, the tool may receive a reference to an encoded text section (e.g., a chapter), again in a format identified by structural metadata. Slightly more complex references include requests such as that to display the *next* or *previous sibling* of a digital object (e.g., next page or previous chapter), or the *parent* of a digital object (e.g., the chapter that includes this page). For images, an intermediary may request the display of a known item at a specific resolution. More challenging will be the standard articulation of a reference to display a

portion (e.g., 250x250 pixels, centered on pixel X.Y) of an image at a specific resolution. Panning, zooming, and cropping of an image are variations on this type of request.

Printing is, we believe, a method similar to the **display** method, differing only in its use of output devices (i.e., printers, plotters, disks, etc.) Indeed, the option to **print** may use the same format as the option to **display**, as is the case with systems relying primarily on encapsulating images in PDF for delivery. The **display** and **print** methods are closely intertwined, and differ based on the formats available from the repository and user preferences. For example, imagine a repository of bitonal, 600dpi page images offers GIF images with interpolated gray-scale, Postscript, and PDF. A user without Acrobat may choose to display the GIF images but print using the Postscript files containing encapsulated images, while another user with Acrobat may choose to rely entirely on the PDF image files so that she might **print** and **display** from the same source.

Combination or Comparison

As the body of materials in the digital library grows, the ability to create combinations of data or perform comparisons becomes increasingly important. “Combining” and “comparing” methods are applicable to both images and text. Most common with art and architectural images (cf. the common use of two slide projectors in art historical instruction), we also frequently see the need to be able to support the comparison of two passages of text, or the display of a text alongside an image. Common applications of this method we might expect to see include:

- synchronous scrolling of a text in two different languages, or a text with commentary;
- the side-by-side display of a text and an image (cf. Rosetti’s paintings and poetry, often using the same title or treating the same theme);
- the display of two images side-by-side;
- the display of an indeterminate number of image objects positioned in a grid;
- the display of a number of image objects positioned in a grid of specific dimensions (e.g., three columns by five rows).

Of somewhat greater complexity is the similar requirement to combine objects. Particularly important here is the need to be able to apply and remove layers on an object. This problem is considerably easier where a single repository has provided these layers with the base image; however, we must define a more generic method to allow the combination of layers from diverse sources. For example, the highly allegorical drawings found in 19th century journals such as *Harper’s Weekly* frequently contain contemporary public persons portrayed as historical figures. A server at one DLF institution might provide the images of the pages, while a second DLF institution might overlay commentary identifying each of the persons. Recall that a tool supporting the display method might offer that image in any number of different resolutions, or even portions of the image cropped and enlarged. The ability to coordinate the annotations of the second institution with the page image from the first institution will require carefully controlled metadata about coordinates that stay constant with the cropped portion. While this sort of administrative and structural metadata may be too challenging for early portions of the project, the metadata model must be flexible enough to accommodate this information in later iterations.

Repository Search⁶

More than most other methods, those for repository search currently tend to be methods exhibited by server-side programs rather than client-side tools. We can easily imagine at least portions of these methods migrating to the user's desktop, inherent in tools for managing and interpreting results, but considerable standardization will first need to take place. In repository search, an intermediary will collect information about the user's query, the characteristics of the available collections, and will begin to process results (e.g., in a sorted list by object or by collection). This intermediary has clearly distinct discovery and retrieval functions, so these will be treated separately below.

In order to support **discovery** within or among repositories, each collection must participate in a *conversation* with the client or intermediary; this conversation constitutes the method we associate with **discovery**. Among the characteristics of a repository's **discovery** method will necessarily be a means to understand the search parameters of the repository, including gathering information on searchable fields, the sorts of operators that can be applied, and other constraints. Of course these mechanisms have been specified in protocols such as Z39.50, but it is important to explore "lighter weight" and *more flexible* mechanisms such as the *SIL* specified by Nigel Kerr's "Personal Collections and Cross-Collection Technical White Paper" (<http://dns.hti.umich.edu/~nigelk/work/pccc.html>).

In order to support the **retrieval** of results within and among repositories, the conversation identified above must also include a well-specified retrieval method. Results must come from the repository or repositories in a well-articulated and easily parsed syntax. The tool will use this syntax, for example, to build result lists, to bring together results from multiple repositories, and to compile results from multiple repositories. An example of a proposed specification for such a retrieval method can also be found in Nigel Kerr's "Personal Collections and Cross-Collection Technical White Paper" (<http://dns.hti.umich.edu/~nigelk/work/pccc.html>).

Color Analysis

An admittedly challenging set of methods will come with richer and more reliable color metadata. The availability of a Color Look Up Table (CLUT), providing color, shape, and texture distribution that can be processed through automatic means, will aid in a variety of tasks. For example, a *color-matching* behavior might take CLUT information from manuscript fragments, locating fragments by color or texture that are more likely to be from the same paper stock. CLUT information can also be used to measure subtle variations in information such as shape and patterns, and thus hidden features such as characters obscured by palimpsest erasures. Methods using the CLUT will support these types of analysis.

⁶ The MoA II effort will rely primarily on a union catalog to effect discovery. A union catalog obviates problems associated with inter-repository searches—how we characterize the search to the various systems, and how we bring together results from those different collections. Nevertheless, the ability to perform a search across a number of distributed repositories becomes increasingly important as we distribute responsibility and sustain important elements of institutional autonomy. Repository search, and especially the means to support *inter-repository* search, is discussed here not so much for the reason that it need be explored in MoA II, but because it is an important method to keep in mind as we conceptualize the digital library.

Bookmarks, Annotation, and Links

As the digital library grows in maturity and capability, the array of interactions with objects will grow more complex. Intra-object bookmarks, annotations, and more sophisticated linking methods are all parts of methods that our users desire; none of these methods is outside the capabilities of readily available desktop technology today. Nevertheless, our ability to support these methods is hampered by unreliable or incomplete metadata, by the absence of generalized notions of user authentication and authorization, and by a lack of support by repositories. Importantly, though, we lack *tools* to exploit methods in these areas. Many of these methods are explored in great detail in the research applications developed by Robert Wilensky and his team as they pursue notions of “multivalent documents.”⁷ Moreover, emerging standards such as the XML linking language (XLL) will help articulate the language for complex links such as a span of information or “the third paragraph in the fourth section” within a remote document. These methods will be best supported through the articulation and adoption of architectures within which effective tools can be built, and metadata that documents a full range of digital object features.

MoA II Metadata

Metadata can be in a header, a MARC record, a database, an SGML file, or distributed amongst a variety of locations. An object or repository only needs to be able to reconstitute the metadata and present it to a user or application when requested (discovery, navigation, and administrative functions).

1) Descriptive Metadata

The library community has a long history of developing standards and best practices for descriptive metadata (e.g., MARC, Dublin Core, etc.). Given existing standards and ongoing work within the community to investigate descriptive metadata issues, the MoA II proposal did not focus on this area. Instead, the proposal to the NEH recommended a MoA II testbed that used a union catalog, with MARC records contributed by the participants. The participants will also contribute finding aids encoded to the EAD community standard.

Therefore, the discovery process will consist of users searching the MoA II union catalog of MARC collection level records that will be linked to their corresponding finding aids, and the finding aids will then be linked to the appropriate archival digital library object (e.g., photograph, diary, etc.). Of course, it will also be possible to search the finding aids directly to discover archival library objects.

2) Structural Metadata

The authors of this white paper offer the following definition of our use of the term “structural metadata”:

⁷ Thomas A. Phelps and Robert Wilensky. “Toward Active, Extensible, Networked Documents: Multivalent Architecture and Applications.” In *Proceedings of DL'96*, 1996.

Structural metadata are those metadata that are relevant to *presentation* of the digital object to the user, describing them in terms of:

- 1) **Navigation**, e.g., navigating internal organization and exploring the relationship of a sub-object to other sub-objects (see Tables of structural metadata features for a definition of sub-objects); and
- 2) **Use**, e.g., the format or formats of the objects available for *use*, rather than those formats stored.

The terminology of the digital library is evolving rapidly. In fact, as can be seen quite readily, even avoiding domains where the term is used much differently (e.g., georeferenced data), quotes on the subject show considerable variation in the way that our community uses the term *structural metadata*. Current thinking divides digital library metadata into either three categories (descriptive, administrative, and structural metadata) or into two categories (descriptive and structural metadata, with structural metadata subsuming both administrative and structural metadata). The approach taken here is to separate administrative and structural metadata, which in turn influences the way we refer to structural metadata. A technical term, like any word, is ultimately defined by the way we use the term. Nevertheless, of most importance here is not that we see structural metadata *as separate* from administrative metadata, but rather that we propose the following categories for inclusion in the MoA II architecture.

It is important to note that even though the categories defined here are *presented* in SGML, the data in a repository will not necessarily be *stored* in an encoded form (such as SGML) or in a table. This document does not advocate particular methods for storing data; the authors believe that various approaches will be necessary at the different institutions, and that different approaches may even exist within the same institution. For example, a single institution may store descriptive metadata in USMARC, portions of structural and administrative data in relational tables, and other portions in SGML. The examples provided here are intended only to illustrate the type of data presented in interactions between repositories and intermediaries. We assume that the metadata documented here will be extracted from a metadata management system (or systems) in interactions with intermediaries such as tools. Further, **default** and **inherited** values will be expressed explicitly at the level of the sub-object, even if implicitly associated with the sub-object through the metadata management system.

Structural Metadata Elements and Features Tables

These tables attempt to be comprehensive and are recommendations for the full set of possible structural metadata elements that we think an individual collection may possibly find useful. Some repositories will only use the minimum set of required elements. Other repositories will also use elements that can be derived in an automated fashion. Still others will choose to use elements that are easy to derive. The table: includes both minimal and maximal values; identifies required and repeatable fields; and identifies which field values may be inherited or supplied manually. Some elements can fulfill both administrative and structural functions.

Some elements are more relevant to “raw” data (such as page scans) that has not required much intellectual examination of the data structure. Other elements are more relevant to “seared” data (such as chapter divisions and headings) that involves only minimal examination of data structure to generate appropriate metadata. Still other elements

are primarily relevant to “cooked” data (such as SGML marked-up text) that has a very involved intellectual examination of its structure.⁸

The table presented here is divided into two primary categories: structural metadata defining the “object,” and structural metadata defining the digital sub-objects themselves (e.g., the individual digital pages). In making this distinction, we divide the structural information for a digital object into that which refers to the constituent objects that cohere into a whole (e.g., a description of the extent of a digital book), and that which is specific to the individual parts (e.g., page or image references). The model presented here owes a great deal to the “[Structural Metadata Dictionary for LC Repository Digital Objects](#),” both for its organization and for many of the elements themselves.

In making the distinction between object and sub-object metadata, we acknowledge that the distinction is in some ways artificial. For example, a tool might assemble information relating to the constituent parts of a photo album by querying each of the constituent sub-objects rather than querying a specially designed digital object. We believe, however, that certain economies prevail when storing information such as ownership (i.e., of the digital object) only once in the object rather than with each sub-object, and this model strives to balance the specification of elements accordingly.

Object Level -- Structural Metadata Elements

Element	Comments	Required	Repeatable
Unique identifier reference	A unique identifier must be presented with each digital resource. In order to ensure the effective coordination of metadata, structural metadata must contain a unique identifier reference (referring to the object’s unique identifier). This unique identifier is intended to be a precursor (and functional equivalent) to an URN.	Yes	No
Content type	Describes the types of content available (i.e., not necessarily formats captured) for this digital object. It consists of paired attributes containing (1) a generic and controlled descriptor of the type of material (e.g., text, page image, continuous tone image, audio, etc.), and (2) the format available for each. Objects will be available in a number of formats. For example: Page: TIFF, GIF, JPEG, PDF Image: JFIF, JTIP, etc. Text: HTML, XML Computationally, a tool or service <i>should</i> be able to do something like say “there are 324 images available in four different formats, thus we can build a table of views that’s four wide and 324 high.” It may not appear as a <i>table</i> , per se, but as options available for selection by a user.	Yes	Yes

⁸ The University of Michigan has been using the terms “raw,” “seared,” and “cooked” to describe levels of processing for the Making of America 1materials (<http://www.umdl.umich.edu/moa/>). For a fuller discussion of MoA1 processing at UM, please see: <http://www.dlib.org/dlib/july97/america/07shaw.html> and various sections in <http://www.umdl.umich.edu/moa/about.html>.

Extent	<p>Extent is in many ways analogous to the extent statement in a MARC record, and for structural data provides information specific to the number and levels of component types. In the simplest terms, this element facilitates the sort of dialogue between object and intermediary that runs as follows:</p> <p>Tool: <i>How are you organized?</i></p> <p>Repository: I have images of pages, 33 sections (at different levels) of Descriptive or structural organization in encoded text, and I have a collection of continuous tone images of photographs that were affixed to the pages.</p> <p>Tool: <i>Okay, let's go for Descriptive organization. Give me all of the section headings because there are fewer than 100.</i></p>	Yes	Yes
---------------	---	-----	-----

Sub-Object Level – Structural Metadata Elements

Feature	Comments	Required	Repeatable	Source
Data file size	The file size of an object sent to an intermediary such as a client or tool. It is distinct from the file size of the object stored or captured, which is administrative data.	No	No	Automatically generated
Structural Division(s) (DIVn)	A digital object may be logically divided into parts (e.g., letters in a diary). If resources are made available to support some level of encoding, structural divisions are encoded with the TEI element DIV n (e.g., <DIV1>). Should an object be encoded for logical features, encoders may wish to exploit other TEI elements (not documented here). The fully encoded document may be offered as a version of the digital object. Many of the attributes of the Digital Object, described below, will be applicable to the Structural Division(s).	No	Yes	Manually supplied in encoding
Sub-object Relationships: Parent, Children, Siblings	Relationships provide information on sub-object relationships. A diary entry in a diary section (e.g., a year) would have as its parent the section, and would have as siblings the previous and next diary entries. If, for example, it was an unusually long diary entry with sections of its own, its “children” would be the sections within the entry.	No	Yes	Automatically generated
Sub-object Type	This category constitutes <i>use</i> information and is intended primarily for pages and includes values such as “TOC1” (i.e., first table of contents page).	No	No	Manually supplied in data capture
Sub-object value	This value carries the page number for objects that are pages. Unnumbered pages may not have an N value, or the N value may be supplied through inference.	No	No	Manually supplied in data capture
Sub-object sequence	Pages require a sequence indicator (e.g., this is the third page in the sequence of pages contained in	Yes	No	Manually supplied in

	this book). Carries a numeric value only and must be specified.			data capture
Sub-object Format	Images of all types (e.g., page images and continuous tone images) require format information. The contents of the Sub-object Format element are coordinated with the Content Type element (see above). While Content Type declares the available formats for a particular “type” of information (e.g., encoded text), the Sub-object Format element refers to these declarations to inform the intermediary of the available formats for the object at hand. For example, a page image may be said to be available as a GIF image, a PDF file, and a TIFF G4 image.	Yes	No	Inherited from Object header
Sub-object dimensions	Dimension information such as the resolution offered by the object (i.e., not the captured resolution) may be provided. This element documents the forms of the image object that can be requested from the repository (i.e., in order to assist an intermediary in navigation, manipulation, etc.). For images of all types (i.e., bitonal and continuous tone), this is resolution and pixel dimensions. The element is not applicable for text. To imagine the relationship between administrative and structural metadata in this regard, we can imagine a repository declaring to a tool: “This image is stored in 1200dpi 24bit color, which is administrative data; it is available to you as 72dpi with a 256 color adaptive palette, which is structural metadata.”	No	No	Inherited from Object header
Sub-object size	This element offers two values, “Reference” and “Full”, and is used to describe generic instances of an image. It is important to note that we favor a perspective that sees all available versions of an image on a continuum going from the lowest usable resolution to a full resolution. However, we acknowledge (1) the “lowest usable resolution” is a moving target and that (2) an <i>arbitrarily</i> specified “reference” version is a useful concept. We would like to propose that the “Reference” version is 500 pixels high (or less, if the original captured was fewer than 500 pixels high). The only other option here is “Full”, which is a 1:1 display of resolution captured.	No	No	Inherited from Object header
Sub-object reference	This attribute carries information needed to locate the sub-object. Where possible, it should be an URN. Alternatively, it must be based on the URN for the object of which the sub-object is a part.	Yes	No	Manually supplied in data capture <i>or</i> automatically generated

3) Administrative Metadata

Administrative metadata consists of the information that allows the repository to manage its digital collection. This includes:

- Data related to the creation of the digital image (date of scan, resolution, etc.);
- Data that can identify an instantiation (version/edition) of the image and help determine what is needed to view or use it (storage or delivery file format, compression scheme, filename/location, etc.) and;
- Ownership, rights, and reproduction information.

Some metadata elements may be both structural and administrative, and may be used for similar purposes in those two areas. (For example, “Content type” is a structural metadata element used to present available file formats to a service, while “File Format” is an administrative metadata element that tells systems administrators what format a particular file is in.)

Administrative metadata is critical for long-term file management. Without well-designed administrative metadata, image file contents may be as unrecognizable and unreadable a decade from now as Wordstar or VisiCalc files are today. Administrative metadata should help future administrators determine the type of file it is, when it was created, what particular original it was created from, what methods or personnel might have introduced artifacts into the image, and where the different parts of this (or related) digital object reside. Eventually, we hope that administrative metadata may help objects care for their own long-term management.

In the past, certain administrative metadata (such as file formats) resided in file headers, while others resided in accompanying databases. At some point in the future, all administrative metadata may reside within the file header, but that would be ineffective until community standards develop on where they would go within the header, how to express them, etc. In this paper we define the administrative metadata fields necessary irrespective of a particular syntax of where these fields will actually reside. And for the purposes of the MOA2 Project, we will deliver all the administrative metadata external to the image file header.

In this section we primarily discuss administrative metadata for “master” files. But in the future, repositories are likely to see “master” files which are themselves derivatives of previous files. In order to make the administrative metadata we identify as compatible as possible with future developments, we have included a minimal amount of information that deals with derivative files (or other instantiations of a work). This will hopefully lay the groundwork for future research projects to be able to identify and trace the provenance of a particular digital work.

Administrative Metadata Elements and Features Tables

As with the structural metadata tables presented above, these tables attempt to be comprehensive and are recommendations for the full set of possible administrative metadata elements that we think an individual collection may possibly find useful. Some repositories will only use the minimum set of required elements. Other repositories will also use elements that can be derived in an automated fashion. Still others will choose to use elements that are easy to derive. The table includes both minimal and maximal values,

which are allowed to repeat, etc. Some elements can fulfill both administrative and structural functions.

Though the number of metadata fields may at first seem daunting, a high proportion of the fields is likely to be the same for all the images scanned during a particular scanning session. For example, metadata about the scanning device, light source, date, etc. is likely to be the same for an entire session. And some metadata about the different parts of a single object (such as the scan of each page of a book) will be the same for that entire object. This kind of repeating metadata will not require keyboarding each individual metadata field for each digital image; instead, these can be handled either through inheritance or by batch-loading of various metadata fields. In any case, this is an attempt to identify best practices for metadata development, and we expect that individual repositories will follow this to the extent that they can afford.

The rest of this section is divided into 4 convenient parts: elements for the creation of a digital master image; identifying the digital image and what is needed to view or use it; linking the parts of a digital object or its instantiations, providing context; and ownership, rights, and reproduction information. The first two parts, both recorded at the point of capture, uniquely identify a particular representation of a work. For future derivative images, these could be iteratively nested to represent the provenance of a work.

Elements for the creation of a digital master image (recorded at point of capture)

Element	Examples	Comments	Required	Repeatable	Source
source type	Photographic print, slide, manuscript, printed page(s), another digital image	to identify the material from which the digital file was created - the item on hand, even if it itself is a reformatted version, e.g. the scan of a 35mm slide of a painting would be entered here as a 35mm slide	Yes	No	
source physical dimensions	10.2cm x 18.4cm	Actual physical dimension scanned if cropped. Needed for appropriate facsimile output	Yes	No	
source characteristics	film type/ASA/manufacturer, print type; tightly bound volume	Relevant if an intermediary stage is actually removed from 'original'; or may add descriptive details of source which may impact on scanning quality	No	No	
source ID	a local catalog unique ID for a book; an accession number for a special collections item	the source of the source image (recursively)	Yes	No	
scanning date	any usual time stamp:	Need date to year/month	Yes	No	Autom

	yyyy/mm/dd or yyyy/mm	specificity to assist in later evaluation of technology at the time of digital master creation.			generat
ICC scanner profile		Describes the color artifacts introduced by the scanning device. Necessary to map the images into standard color space and to adjust for display and printing devices.	No	No	Genera session
Light source	Example: 3400K Tungsten, infrared, Osram Delux L fluorescent	Should be specific to settings for this scan (f-stop, electronic shutter speed, filtering, illumination level); may be necessary in later evaluation of color capture. Again, may be specific to each image or by inheritance to collections of images a via a separate descriptive file (with anomalies indicated per image as needed).	No	No	Batch g
Resolution	600 dpi; 400 dpi interpolated to 600 dpi; 1536 x 1024	the settings on the input scanning device (cameras usually measure these in dimensions, other devices in dpi). Note where device does its own interpolation.	No	No	Autom generat

Identifying the digital image (master or derivative) and what is needed to view or use it (recorded at point of capture).

This metadata is needed by applications programs in order to recognize how to initially display the digital object, including notes as to file format, compression schemes, and color space. Today there are common conventions for carrying much of this information in a file's header, but it is not certain that future software will be able to interpret all of these. (For example, today's operating systems can look at the beginning of file headers and determine whether the file is Microsoft Word, Adobe Acrobat, or a JFIF/JPEG file. But those same operating systems may not recognize less common file formats, or even earlier versions of a format as prevalent as Microsoft Word.) It is important that this information be expressed in a consistent way in fields within the file header that are clearly readable (such as uncompressed ASCII). There are common conventions for a number of these fields, as well as for expression of field contents. In addition, applications software automatically saves most of this information within the

header when files are created, so for most of these fields there is no issue of expense or difficulty in finding and noting the proper metadata.

Element	Examples	Comments	Required	Repeatable	Source
Type of Image	MIME TYPES, bit-mapped	To determine what class of image, and hence what general viewer type will be needed.	Yes	No	Autom generated
File format	TIFF, GIF, JFIF, SPIFF, FlashPix, JTIP, PICT, PCD, PhotoShop, EPS,	File format needed for viewer to display image.	Yes	No	Autom generated
(Lossless) compression format	LZW	Type of algorithm needed to decompress the image, with note of software package used to apply the format, and degree/percentage of compression used where options exist.	Yes	No	Autom generated
Dimensions	310 x 437	Dimensions in pixels, often needed by viewer, and acts as an indication of quality to user.	Yes	No	Autom generated
Bit-depth	1, 8, 24, color, grey scale	Color depth, often needed by viewer and acts as an indication of quality to user.	Yes	No	Autom generated
Color Lookup Table (CLUT)	(usually a binary table of RGB values)	Color values actually used in the image, often needed by some file formats, especially GIF.	Yes	No	Autom generated
Color space	CMYK, RGB, Lab	Color space used, often needed by viewer and indicates whether image was initially created for onscreen display or for pre-press output. (Some color space parameters such as white point may require individual tags).	Yes	No	Autom generated

Linking the parts of a digital object or its instantiations, providing context (overlaps with Structural)

Context metadata

Element	Examples/Definitions	Comments	Required	Repeatable	Source
Overall View Image	the image file representing the overall	If this image file represents a detail or part	No	No	

	view (used to find the location of a detail within the overall image); may not apply if details are not used	of a another image, the parent file should be indicated here.			
Sequence Number	Example: 4 of 5; 2B of [Primary Image ID](?)	Relative position of a particular image in an image file chain that begins with the file named in the PRIMARY IMAGE tag. There is no one clear-cut labeling system for this concept: should an arbitrary sequence number be provided for detail images that are linked to a primary image when there is no obvious sequence? Unlike the pages of a book, where sequencing is critical to textual integrity, detail images of a larger view image may be sequenced according to many different criteria, i.e. wide view to narrow view, left to right, top to bottom, or no particular order at all.	No	No	
Sequence Total	250 (equal to the number of related image files in the chain; i.e. the book has 250 pages, each page represented by one image file)	The total number of image files in a given sequence; this number may not be known at the time of capture but should be 'calculated' prior to use of master or beginning of derivative production.	No	No	
Version (from MESL Data Dictionary - NR)	10 /13/95; from the second edition of the permanent catalog	This field contains the full text of any information that the content provider considers necessary to uniquely identify the version of this information represented. This field may contain an arbitrary number or the date of creation of the electronic data set, or	No	No	

		may point to any internal version control information needed by the content provider.			
Version Date	DEFINITION: mm/dd/yy.	date the version referred to in VERSION was created; could be somewhat redundant if this data is used in the VERSION field	No	No	

Ownership, rights, and reproduction information

Element	Examples/Definitions	Comments	Required	Repeatable	Source
Owner	EXAMPLE: Saskia	Owner(s) of the copyright on the digital image file, which MAY be the creator of the digital image file, or the person(s) from whom the digital image file was purchased or licensed. It should contain the name(s) of the person(s) from whom copy/distribution and display/transmission rights may be secured. Note: this refers to the copyright on the digital image only, not the work(s) represented in the digital image.	No	Yes	Likely for an e collect
Owner Number NR)	Example: [widely varies]	Any number or alphanumeric string that uniquely identifies the image as belonging to the owner. It can take the form of numbers, text strings, barcodes, electronic ID number, or file name.	No	No	Likely for an e collect
Copyright Date	EXAMPLE: 1997	Date of copyright expressed as yyyy; in current approaches to interpretation of copyright law, the year is sufficient information.	No	No	Likely for an e collect
Credit Line (from MESL data Dictionary -	DEFINITION: The text required to be displayed		No	No	Likely for an e collect

NR)	whenever the image/data appears. EXAMPLE: Copyright Berkeley Art Museum, 1978. All rights reserved.				collect
Copy / Distribution Restrictions	DEFINITION: text that spells out any copyright restrictions pertaining to the copy and distribution of this image file. EXAMPLE: Copy and distribution of this file is prohibited without the express written consent of...		No	No	Likely for an e collect
Display/Transmission Restrictions	DEFINITION: text that spells out any copyright restrictions regarding the transmission and display of this image file. EXAMPLE: This file may be displayed or transmitted across a network only by person(s) who have signed a license agreement with ...		No	No	Likely for an e collect
License Term	DEFINITION: specifies the duration of any licensing arrangement covering this image		No	No	Likely for an e collect
License Begin Date	DEFINITION: start date of any licensing agreement covering this image expressed as mm/dd/yy		No	No	Likely for an e collect
License End Date	DEFINITION: end date of any licensing agreement covering this image expressed as mm/dd/yy		No	No	Likely for an e collect

Encoding – Best Practices

Encoding Archival Object Content and Finding Aids

Many of the MoA II materials will involve text encoding. This may be the case whether the documents are carefully transcribed and edited versions of the original documents, whether they simply organize (conceptually) a mass of automatically generated OCR, or whether only the framework of a document is encoded, with pointers to images. Moreover,

finding aids for many resources will be encoded to support fine-grained access to a collection. The DLF is fortunate to be able to rely on substantial work and large community efforts in both of these areas. Moreover, work is underway in the DLF to organize discussions around our use of the available guidelines.

For the encoding of finding aids, the Encoded Archival Description (EAD) should be used by project participants. Information about the EAD, including guidelines for the application of the EAD as well as DTDs, is available at <http://lcweb.loc.gov/ead/>. While it is the case that the EAD guidelines allow considerable latitude for the application of markup to finding aids, work with the EAD must grow out of local assessment of the needs for finding aid support and the way that these finding aids will be used. Discussions are underway in the DLF surrounding inter-institutional searching of EAD-encoded collections, and the ways that this will cause us to give scrutiny to local practice. Rather than those proposed efforts, continuing to drive the application of EAD by locally defined needs will help clarify the range of needs for inter-institutional applications.

Text encoding efforts in MoA II will be well supported by the SGML articulated in the Text Encoding Initiative Guidelines (TEI). Information about the TEI can be found at: <http://www-tei.uic.edu/orgs/tei/>. A searchable/browsable version of the TEI Guidelines can be found at: <http://www.hti.umich.edu/docs/TEI/>. The TEI offers support for a broad range of types of documents and methods, including the transcription of primary sources and damaged documents. More importantly, the TEI Guidelines and the associated DTDs offer support for encoding the wide range of structures that may be present in MoA II documents, regardless whether transcriptions are included. Just as with the EAD, the TEI offers considerable flexibility in the ways documents can be encoded. Discussions are underway in the DLF surrounding a June/July meeting to discuss the use of the TEI by digital library projects.

A further note on the relevance of XML to these two central encoding schemas may be useful. XML promises to bring richly encoded documents to the user's desktop through widely available browsers. Moreover, a growing array of XML-capable tools should be available through mainstream software development. It is the expectation of the authors of this white paper that we will soon see XML-compliant versions of both the TEI and the EAD DTDs. One editor of the TEI Guidelines has been centrally involved in the writing of the XML specifications and the TEI editors have declared their intention to create XML-compliant versions of the widely used TEI DTDs. We are likely to see no less from the EAD.

Encoding to Encapsulate Metadata and Content inside the Archival Object

After this White Paper is circulated and discussed, the MoA II project team will have gathered enough information to define an encoding scheme for the archival objects that will populate the MoA II testbed. The team will develop an XML DTD that will be used to encode these objects. This XML DTD will then define the transfer syntax used for MoA II objects. Selecting XML to encode the object does **not** mean that any repository must use that encoding for internal object storage. However, this does give DLF and the larger community an opportunity to discuss and evaluate XML as transfer syntax.

Part IV: Best Practices for Image Capture

For the Making of America project, the participants need to have some common practices to follow. The “best practices” outlined in this document are most relevant to the classes of archival materials chosen for this project, but many of them are applicable to other classes of physical objects as well.

Because image capture capabilities are changing so rapidly, we will divide the “best practices” discussion into two parts: general recommendations that should apply to many different types of objects, and specific minimum recommendations to be used during the course of the MoA II testbed project. Below you will find a discussion of the practices that we think are fairly universal, and we believe that this portion of the document will be usable for many years to come. This includes the notion of masters and derivatives, when images should be corrected to “look better”, etc. This also contains some discussion of how to go about selecting image quality for a particular collection, and issues in choosing file formats. The section, Summary of General Recommendations, found at the end of Part IV provides a list of these suggested best practices.

But the issues of image quality and file formats are both complex (and vary from collection to collection) and in flux (due to rapid technological developments and emerging standards). Therefore, at the end of Part IV, we have also summarized the more specific recommendations to be employed right now in MoA II, and provide a list of minimally acceptable levels rather than a precise set of guidelines (see: Specific Minimum Recommendations for this Project).

Recommendations for the full set of structural and administrative metadata are listed in Part III (above). Standards and procedures for the image capture process are described below.

Scanning

Appropriate scanning procedures are dictated by the nature of the material and the product one wishes to create. There is no single set of image quality parameters that should be applied to all documents that will be scanned. Decisions as to image quality typically take into consideration the research needs of users (and potential users), the types of uses that might be made of that material, as well as the artifactual nature of the material itself. The best situation is one where the source materials and project goals dictate the image quality settings and the hardware and software one employs. Excellent sources of information are available, including the experience of past and current library and archival projects (see Appendix Bibliography section entitled “Scanning and Image Capture”). The pure mechanics of scanning are discussed in Besser (Procedures and Practices for Scanning), Besser and Trant (Introduction to Imaging) and Kenney’s Cornell manual (Digital Imaging for Libraries and Archives). It is recommended that imaging projects consult these sources to determine appropriate options for image capture. Decisions of quality appropriate for any particular project should be based on best anticipation of use of the digital resource.

Digital Masters And Their Derivatives

Digital master files are created as the direct result of image capture. The digital master should represent as accurately as possible the visual information in the original object. (Note: this is similarity in terms of technical fidelity using objective technological measurements; this is not accuracy as determined by comparing the object scanned to an image on a display device. This type of accuracy is obtained by the manipulation of settings before scanning rather than by image processing after scanning). The primary functions of digital master files are to serve as an archival image and as a source for derivative files. In the archival sense, a digital master file may serve as a surrogate for the original, may completely replace originals or be used as security against possible loss of originals due to disaster, theft and/or deterioration. Derivative files are created from digital master images for editing or enhancement, conversion of the master to different formats, and presentation and transmission over networks. Typically, one would capture the master file at a very high level of image quality, then would use image processing techniques (such as compression and resolution reduction) to create the derivative images which would be delivered to users.

Long term preservation of digital master files requires a strategy of identification, storage, and migration to new media and policies about their use and access to them. The specifications for derivative files used for image presentation may change over time; digital masters with an archival purpose can be processed by different presentation methods to create necessary derivative files without the expense of digitizing the original object again.

Image Quality

Image quality for digital capture from library originals is a measure of the completeness and the accuracy of the capture of the visual information in the original. There is some subjectivity involved in determining completeness and accuracy (should the digital representation of faded or stained handwriting show legibility or reflect the illegibility of the source material?). Image quality should be judged in terms of the goals of the project.

Image quality depends on the project's planning choices and implementation. Project designers need to consider what standard practices they will follow for input resolution and bit depth, layout and cropping, image capture metric (including color management), and the particular features of the capture device and its software. Benchmarking quality (see Kenney's Cornell Manual) for any given type of source material can help one select appropriate image quality parameters that capture just the amount of information needed from the source material for eventual use and display. By maximizing the image quality of the digital master files, managers can ensure the on-going value of their efforts, and ease the process of derivative file production.

Quality is necessarily limited by the size of the digital image file, which places an upper limit on the amount of information that can be stored. The size of a digital image file depends on the size of the original and the resolution of capture (number of pixels in both height and width that are sampled from the original to create the digital image), the number of channels (typically 3: Red, Green, and Blue: "RGB"), and the bit depth (the number of data bits used to store the image data for one pixel).

Measuring the accuracy of visual information in digital form implies the existence of a capture metric, i.e., the rules that give meaning to the numerical data in the digital image

file. For example, the visual meaning of the pixel data Red=246, Green=238, Blue=80 will be a shade of yellow, which can be defined in terms of visual measurements. Most capture devices capture in RGB using software based on the video standards defined international agreements. A useful introduction to these topics can be found in Poynton's Color FAQ: <<http://www.inforamp.net/~poynton/ColorFAQ.html>>. We strongly urge that imaging projects adopt standard target values for color metrics as Poynton discusses, so that the project image files are captured uniformly.

A reasonably well-calibrated grayscale is all but required for measuring and adjusting the capture metric of a scanner or digital camera. We recommend that a standard target consisting of grayscale, centimeter scale, and standard color patches be included along one edge of every image captured, to provide an internal reference within the image for linear scale and capture metric information. Kodak makes a set consisting of grayscale (with approximate densities), color patches, and linear scale which is available in two sizes: 8 inches long (Q-13, CAT 152 7654) and 14 inches long (Q-14, CAT 152 7662)

Bit depth is an indication of an image's tonal qualities. Bit depth is the number of bits of color data which are stored for each pixel; the greater the bit depth, the greater the number of gray scale or color tones that can be represented and the larger the file size. The most common bit depths are:

- Bitonal or binary, 1 bit per pixel; a pixel is either black or white
- 8 bit gray scale,; 8 bits per pixel; a pixel can be one of 256 shades of gray
- 8 bit color, 8 bits per pixel ("paletted color"); a pixel is one of 256 colors
- 24 bit color (RGB), 24 bits per pixel; each 8-bit color channel can have 256 levels, for a total of 16 million different color combinations

While it is desirable to be able to capture images at bit depths greater than 24 (which only allows 256 levels for each color channel), standard formats for storing and exchanging higher bit-depth files have not yet evolved, so that we expect that (at least for the next few years) the majority of digital master files will be 24-bit. Project planners considering bitonal capture should run some samples from their original materials to verify that the information captured is satisfactory. 8-bit color is almost never suitable for digital masters.

Lossy compression is unwise, as we do not yet know how today's lossy compression schemes (optimized for human eyes viewing a CRT screen) may affect future uses of digital images (such as computer-based analysis systems or display on future display devices). Unlike lossy compression, lossless compression will not eliminate data we may later find useful. But lossless compression adds a level of complexity to decoding the file many years hence. And many vendor products that claim to be lossless (primarily those that claim "lossless JPEG") are actually lossy. Those who choose lossless compression should make sure they take into consideration digital longevity issues.

The objective of color management is to control the capture and reproduction of color in such a way that an original print can be scanned, displayed on a computer monitor, and printed, with the least possible change of appearance from the original to the monitor to the printed version. This objective is made difficult by the limits of color reproduction: input devices such as scanners cannot "see" all the colors of human vision, and output devices such as computer monitors and printers have even more limited ranges of colors they can reproduce. Most commercial color management systems are based on the ICC

(International Color Consortium) data interchange standard, and are often integrated with image processing software used in the publishing industry. They work by systematically measuring the color properties of digital input devices and of digital output devices, and then applying compensating corrections to the digital file to optimize the output appearance. Although color management systems are widely used in the publishing industry, there is no consensus yet on standards for how (or whether) color management techniques should be applied to digital master files. Until a clear standard emerges, it is not recommended that digital master files be routinely processed by color management software.

Useful image quality guidelines for different types of source materials are listed in Puglia & Rosinski's NARA Guidelines and in Kenney's Cornell Manual (see bibliography).

Formats

Digital masters should capture information using color rather than grayscale approaches where there is any color information in the original documents. Digital masters should use lossless compression schemes and be stored in internationally recognized formats. TIFF is a widely used format, but there are many types of TIFF files, and consistency in use of the files by a variety of applications (viewers, printers etc.) is a necessary consideration. In the future, we hope that international standardization efforts (such as ISO attempts to define TIFF-IT and SPIFF) will lead vendors to support standards-compliant forms of image storage formats. Proprietary file formats (such as Kodak's Photo CD) should be avoided.

Image Metadata

Metadata or data describing digital images must be associated with each image created, and most of this should be noted at the point of image capture. Image metadata is needed to record information about the scanning process itself, about the storage files that are created, and about the various pieces that might compose a single object.

As mentioned earlier in this paper, the number of metadata fields may at first seem daunting. However, high proportions of these fields are likely to be the same for all the images scanned during a particular scanning session. For example, metadata about the scanning device, light source, date, etc. is likely to be the same for an entire session. And some metadata, about the different parts of a single object (such as the scan of each page of a book), will be the same for that entire object. This kind of repeating metadata will not require keyboarding each individual metadata field for each digital image; instead, these can be handled either through inheritance or by batch-loading of various metadata fields.

Administrative metadata includes a set of fields noting the creation of a digital master image, identifying the digital image and what is needed to view or use it, linking its parts or instantiations to one another, and ownership and reproduction information. Structural metadata includes fields that help one reassemble the parts of an object and navigate through it. Details about administrative and structural metadata tags are noted in Part III.

Summary of General Recommendations

- Think about users (and potential users), uses, and type of material/collection
- Scan at the highest quality that does not exceed the likely potential users/uses/material

- Do not let today's delivery limitations influence your scanning file sizes; understand the difference between digital masters and derivative files used for delivery
- Many documents which appear to be bitonal actually are better represented with grayscale scans
- Include color bar and ruler in the scan
- Use objective measurements to determine scanner settings (do NOT attempt to make the image good on your particular monitor or use image processing to color correct)
- Don't use lossy compression
- Store in a common (standardized) file format
- Capture as much metadata as is reasonably possible (including metadata about the scanning process itself)

Specific Minimum Recommendations for this Project

- Images with color should be captured in 24-bits; images without color should be captured in 8 bits of grayscale
- Most images should be captured at 600 dpi or greater
- Be very careful if you use compression at all
- If you use the TIFF file format, be sure to note (in the metadata) which TIFF version you used

Appendices

Structural Metadata Notes

1. “Structural metadata is metadata that describes the types, versions, relationships and other characteristics of digital materials.” (Arms, Blanchi, Overly; from section “Data types, structural metadata ...”; <http://www.dlib.org/dlib/february97/cnri/02arms1.html>)
2. “Structural metadata [for digital objects for individual versions] ... includes other metadata associated with the specific version. It includes fields for description, owner, handle of meta-object, data size, data type (e.g., “jpg”), version number, description, date deposited, use (e.g., “thumbnail”), and the date of the last revision.” (Arms, Blanchi, Overly; section on “digital objects for individual versions”)
3. “Structural metadata [for the meta-object] is the metadata that applies to the original photograph and to all of its versions. It includes a description, the owner, the number of versions, the date deposited, the use (“meta-object”), and the date of last revision. If bibliographic information were to be included, it would be added to this part of the meta-object.” (Arms, Blanchi, Overly; section on “Meta-object”)
4. “Schema definitions are of course very basic forms of metadata. We refer to a schema definition language as structural metadata, and distinguish it from the representation of semantics, meaning, and purpose—for which we would use the term semantic metadata. In general, we would like a single metadata model to encompass structure and semantics, and, preferably capable of representing most data models.” (Morgenstern; <http://dri.cornell.edu/Public/morgenstern/registry.htm>)
5. “Looking at the larger picture, there are three type of “metadata” which have been identified by the National Digital Library Project of the US Library of Congress as being relevant to digital collections, namely: (1) descriptive metadata (such as MARC cataloguing records, finding aids, or locally developed practices for describing what's the images are about); (2) structural metadata (the information that ties the images to each other to make up a logical unit such as a journal article or archival folder); (3) administrative metadata (what allows the repository to manage the digital collection, such as scan date and resolution, storage format and filename).” (Gartner et al., DRH '97; <http://users.ox.ac.uk/~drh97/Papers/Gartner.html>)
6. “The [METADATA WORKING] Group worked with the broadest definition of metadata; that is, data about data. It was agreed that the purpose of metadata was a) to help the user discover or locate resources; b) to describe those resources in order to help users determine whether the resources would be useful; and c) to provide physical access to the electronic resource. In the broadest terms, metadata can be characterized as either descriptive or structural. Descriptive metadata, such as a MARC record, provides intellectual access to a work while structural metadata, such as a TIFF header, can be queried and operated on to provide physical access and navigational structure to a document-like object. Much of the discussion in the MWG meetings focused on descriptive metadata; however, a subgroup of the MWG and the Full-Text Working Group met to identify and assess the structural and descriptive metadata which underlies the various scanned image projects at Cornell.” (Cornell MDWG; <http://www.library.cornell.edu/DLWG/MWGReporA.htm>)

7. “[Cornell University Library] should embed structural metadata within full-text resources to enable direct access to special document features, such as tables of contents, title pages, indices, etc., and also correlate image sequence numbers to actual page numbers of the document to enhance navigation within “loosely-bound” electronic documents (e.g. individual scanned image files for pages of a document).” (“Distillation of [Cornell UL] Working Group Recommendations”; <http://www.library.cornell.edu/DLWG/DLMtg.html>)
8. “Structural metadata is used for creation and maintenance of the information warehouse. It fully describes information warehouse structure and content. The basic building block of structural metadata is a model that describes its data entities, their characteristics, and how they are related to one another. The way potential information warehouse users currently use, or intend to use, enterprise measures provides insight into how to best serve them from the information warehouse; i.e., what data entities to include and how to aggregate detailed data entities. An Visible Advantage information warehouse data model provides a means of documenting and identifying both strategic and operational uses of enterprise measures. It also provides the capability to document multi-dimensional summarization of detail data.” (Perkins; <http://www.esti.com/iw.htm>)
9. “Structural metadata identifies the system of record for all information warehouse data entities. It also fully describes the integration and transformation logic for moving each information warehouse entity from its system of record to the information warehouse. In addition, structural metadata defines the refreshment schedule and archive requirements for every data entity.” (Perkins, see above)
10. “*structural data* - This is data defining the logical components of complex or compound objects and how to access those components. A simple example is a table of contents for a textual document. A more complex example is the definition of the different source files, subroutines, data definitions in a software suite.” (Lagoze, Lynch, and Daniel; <http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Body/ncstrl.cornell/TR96-1593/html>)

Bibliography

Organization of Information for Digital Objects

The article “An Architecture for Information in Digital Libraries” by William Arms, Christophe Blanchi and Edward Overly of the Corporation for National Research Initiatives and published in **D-Lib Magazine**, February 1997 issue. <http://www.dlib.org/dlib/february97/cnri/02arms1.html>

Repository Access Protocol – Design Draft – Version 0.0 by Christophe Blanchi of CNRI is found at <http://titan.cnri.reston.va.us:8080/pilot/locdesign.html> and begins “ This document describes the repository prototype for the Library of Congress. This design is based on version 1.2 of the Repository Access Protocol (RAP) and the Structural Metadata Version 1.1 from the Library of Congress.”

“The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata” by Carl Lagoze, Digital Library Research Group, Computer Science Department, Cornell University; Clifford A. Lynch, Office of the President, University of California, and Ron Daniel Jr., Advanced Computing Lab, Los Alamos National Laboratory (July, 1996)

<http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Body/ncstr1.cornell/TR96-1593/html>

Metadata

[Cornell University Library] METADATA WORKING GROUP REPORT to Senior [Library] Management, JULY 1996

<http://www.library.cornell.edu/DLWG/MWGReporA.htm>

and the related work “Distillation of [Cornell UL] Working Group Recommendations” November, 1996

<http://www.library.cornell.edu/DLWG/DLMtg.html>

“Information Warehousing: A Strategic Approach to Data Warehouse Development” by Alan Perkins, Managing Principal of Visible Systems Corporation (White Paper Series)

<http://www.esti.com/iw.htm>

SGML as Metadata: Theory and Practice in the Digital Library. Session organized by Richard Gartner (Bodleian Library, Oxford)

<http://users.ox.ac.uk/~drh97/Papers/Gartner.html>

“A Framework for Extensible Metadata Registries” by Matthew Morgenstern of Xerox, a visiting fellow of the Design Research Institute at Cornell

<http://dri.cornell.edu/Public/morgenstern/registry.htm>

Using the Library of Congress Repository model, developed and used in the National Digital Library Program:

The Structural Metadata Dictionary for LC Repository Digital Objects

<http://lcweb.loc.gov:8081/ndlint/repository/structmeta.html>

which then leads to further documentation of their Data Attributes

<http://lcweb.loc.gov:8081/ndlint/repository/attribs.html>

with a list of the attributes

<http://lcweb.loc.gov:8081/ndlint/repository/attlist.html>

and their definitions

<http://lcweb.loc.gov:8081/ndlint/repository/attdefs.html>

The same site then gives examples of using this model for a photo collection

<http://lcweb.loc.gov:8081/ndlint/repository/photo-samp.html>

a collection of scanned page images

<http://lcweb.loc.gov:8081/ndlint/repository/timag-samp.html>

and a collection of scanned page images and SGML encoded, machine-readable text

<http://lcweb.loc.gov:8081/ndlint/repository/sgml-samp.html>

Scanning And Image Capture

Howard Besser and Jennifer Trant. Introduction to Imaging. Getty Art History Information Project.

http://www.gii.getty.edu/intro_imaging/

Image Quality Working Group of ArchivesCom, a joint Libraries/AcIS Committee. Technical Recommendation for Digital Imaging Projects,

<http://www.columbia.edu/acis/dl/imagespec.html>

Howard Besser. Procedures & Practices for Scanning, Procedures and Processes for Scanning. Canadian Heritage Information Network (CHIN),

<http://sunsite.Berkeley.edu/Imaging/Databases/Scanning>

Electronic Text Center at Alderman Library, University of Virginia. "Image Scanning: A Basic Helpsheet,

<http://etext.lib.virginia.edu/helpsheets/scanimage.html>

Electronic Text Center at Alderman Library, University of Virginia. "Text Scanning: A Basic Helpsheet"

<http://etext.lib.virginia.edu/helpsheets/scantext.html>

Michael Ester. Digital Image Collections: Issues and Practice. Washington, D.C. , Commission on Preservation and Access (December, 1996).

Carl Fleischhauer. Digital Historical Collections: Types, Elements, and Construction. National Digital Library Program, Library of Congress,

<http://lcweb2.loc.gov/ammem/elements.html>.

Carl Fleischhauer. Digital Formats for Content Reproductions. National Digital Library Program, Library of Congress.

<http://lcweb2.loc.gov/ammem/formats.html>

Picture Elements, Inc. Guidelines for Electronic Preservation of Visual Materials (revisiosn 1.1, 2 March 1995). Report submitted to the Library of Congress, Preservation Directorate.

Reilly, James M and Franziska S. Frey, "Recommendations for the Evaluation of Digital Images Produced from Photographic, Microphotographic, and Various Paper Formats" Report to the Library of Congress, National Digital Library Project by Image Permanence Institute. May, 1996
<http://lcweb2.loc.gov/ammem/ipirpt.html>

Anne R. Kenney. Digital Imaging for Libraries and Archives. Cornell University Library, June 1996.

International Color Consortium:
<http://color.org/>

Steven Puglia and Barry Roginski. NARA Guidelines for Digitizing Archival Materials for Electronic Access, College Park: National Archives and Records Administration, January 1998.
<http://www.nara.gov/nara/vision/eap/digguide.pdf>

International Organization for Standardization, Technical Committee 130 (n.d.). ISO/FDIS 12639: Graphic technology – Prepress digital data exchange – Tag image file format for image technology (TIFF/IT). Geneva: International Organization for Standardization (ISO).

Poynton's Color FAQ:
<http://www.inforamp.net/~poynton/ColorFAQ.html>