The Making of America II Testbed Project:

A Digital Library Service Model

December 1999

by Bernard J. Hurley
John Price-Wilkin
Merrilee Proffitt
Howard Besser

ISBN 1-887334-72-6

Published by:

The Digital Library Federation
Council on Library and Information Resources
1755 Massachusetts Avenue, NW, Suite 500
Washington, DC 20036

Additional copies are available for $15.00 from the above address. Orders must be prepaid, with checks made payable to the Council on Library and Information Resources.

The paper in this publication meets the minimum requirements of the American National Standard for Information Sciences—Permanence of Paper for Printed Library Materials ANSI Z39.48-1984.

The Digital Library Federation

On May 1, 1995, 16 institutions created the Digital Library Federation (additional partners have since joined the original 16). The DLF partners have committed themselves to "bring together—from across the nation and beyond—digitized materials that will be made accessible to students, scholars, and citizens everywhere." If they are to succeed in reaching their goals, all DLF participants realize that they must act quickly to build the infrastructure and the institutional capacity to sustain digital libraries. In support of DLF participants' efforts to these ends, DLF launched this publication series in 1999 to highlight and disseminate critical work.

About the Authors

Bernard J. Hurley is the Berkeley Library's director of library technologies. He formerly served as the director for library systems, beginning in 1981, and has worked in the field of library automation for the last seventeen years. While at Berkeley, he has played a central role in developing the GLADIS System, Berkeley's online catalog, catalog maintenance, authority control, and circulation system, and its access to the Berkeley Campus Information Network.

John Price-Wilkin is head of Digital Library Production Services (DLPS) at the University of Michigan, a position he has held since its inception in 1996. A federated organization, DLPS is responsible for mounting and maintaining licensed content for the University of Michigan and several other universities, and for text conversion projects such as the Making of America. Among the units in the DLPS is the University of Michigan's Humanities Text Initiative, an organization responsible for SGML document creation and online systems, which Price-Wilkin founded in 1994.

Merrilee Proffitt is director of digital archive development at the Bancroft Library, University of California at Berkeley. She has worked for the Berkeley Library since 1988. Ms. Proffitt has a broad range of experience in digital library projects, especially those relating to special collections. She has expertise in database development, text encoding, and metadata, and has managed a range of digital imaging and text encoding projects.

Howard Besser is an associate professor at UCLA's School of Education and Information, where he teaches and does research on multimedia, image databases, digital libraries, Web design, information literacy, distance learning, intellectual property, and the social and cultural impact of new information technologies. Besser has been involved with the Dublin Core metadata standard since its inception, and has recently been promoting the extension of metadata activities beyond the "discovery" orientation of the Dublin Core. He was an organizer of a DLF / NISO meeting to develop metadata standards for digital images, and was member of the metadata committee of the US / European Community Digital Collaboratory, laying out directions for future digital library research.

Acknowledgments

No work is produced in a vacuum. The authors would like to thank the following individuals who offered guidance and wisdom to this project: Caroline Arms, Rick Beaubien, Robert DeCandido, Dale Flecker, Rebecca Graham, Susan Hamburger, Bonnie Hardwick, John Hassan, Peter Hirtle, Tim Hoyer, Dan Johnston, Sue Kellerman, Heike Kordish, Steven Mandeville-Gamble, Jerome McDonough, Ralph Moon, Sue Rosenblatt, Kathlin Smith, Christie Stephenson, and Ann Swartzell.

Many thanks are also due to the institutions that participated in the MoA II project. Without the expert staff working on the project, none of this would have been possible.

Finally, we owe a debt of gratitude to the organizations that funded the MoA II project itself. Thanks are due to the Digital Library Federation, for funding the research phase of the MoA II project, and to the National Endowment for the Humanities for funding the testbed phase of the project.

Contents

Figures and Tables

Foreword

Metadata is what makes it possible to locate, provide access to, navigate, and manage digital information in diverse forms. Ongoing work on metadata definition has been critical to the development of digital libraries. The extension and refinement of the Dublin Core, efforts to establish a set of technical metadata elements for images, and other initiatives are expanding the application and usefulness of metadata. Echoing earlier published works, this paper emphasizes the importance of metadata in those developments.

The work of the Making of America II Testbed Project reported in this paper represents a singular effort in digital library development to find ways to provide access to and navigate a variety of materials. In this endeavor, a digital library service model has been defined that encapsulates the interaction of digital objects (including their metadata), tools, and services based on principles of object-oriented design. In developing the digital library service model, project participants did extensive work to identify and define the structural and administrative (often referred to as technical) metadata elements that are crucial in the development of the digital library services and tools.

The Digital Library Federation's support of this work was driven by two of its program priorities: to stimulate the development of a core digital library infrastructure and to organize, provide access to, and preserve knowledge. This publication—DLF's third—furthers the interests of the Federation and its members by presenting one possible model of digital library development for review and discussion within the DLF community and the digital library community at large.

Rebecca Graham

Reader's Guide

Drawing on the example of the Making of America II Testbed Project, this report examines an object-oriented approach to digital library construction, the collection of structural and administrative metadata, and the development of tools to assist scholars. It is divided into four main parts. Readers should approach the report part by part, focusing on those areas of particular interest.

- The *Executive Summary* provides an overview of the MoA II Testbed Project and describes the content and objectives of this report.
- Part I, *Project Background,* describes the history of the project and outlines the activities to be undertaken during each of the three phases.
- Part II, *The MoA II Digital Library Service Model,* reviews the technical details of the model for digital library objects. It briefly describes the three layers of the project: services, tools, and digital library objects.
- Part III, *Implementing the Service Model,* is the most detailed section of the report. It discusses the use of tools in the digital library, presents an overview of structural and administrative metadata, and provides recommendations for the collection of metadata.

Recommendations for imaging are not covered in this report. This topic will be covered extensively in *Guides to Quality in Visual Resource Imaging,* which the Council on Library and Information Resources and The Research Libraries Group will publish on the Web in early 2000.

EXECUTIVE SUMMARY

The Making of America Testbed Project, coordinated by the Digital Library Federation (DLF), is a multiphase endeavor. Its purpose is to investigate important issues in the creation of an integrated but distributed digital library of archival materials (that is, digitized surrogates of primary source materials found in archives and special collections). Drafted during the MoA II planning phase, this report identifies a starting point for the testbed that is being created in the production phase of this project, which is funded by the National Endowment for Humanities.

The library community has a distinguished history of developing standards to enhance the discovery and sharing of print materials: they include, for example, MARC, Z39.50, and interlibrary loan protocols. This leadership continues today, as libraries create new best practices and standards that address digital collections and content issues. The primary goal of this report is to open a dialogue about digital library standards, specifically, to discuss any new best practices and standards that will be required to enable the digital library to meet traditional collection, preservation, and access objectives.

This report asks the question, "How can we create integrated digital library services that operate across multiple, distributed repositories?" Existing standards and best practices clearly play an important role in answering this question. However, this report and the MoA II Testbed Project raise a new area of discussion that goes beyond the discovery of a digital object and address how it is handled. The report and the testbed focus on the need to develop standards for creating and encoding digital representations of archival objects (for example, a digitized photograph or a digital representation of a book or diary). If tools are to be developed that work with digitized archival objects across distributed repositories, these objects will require some form of standardization.

This report begins the discussion of digital object definitions by developing and examining metadata standards for digital representations of a variety of archival objects, including text, digitized page images, photographs, and other forms. For the purposes of this report, there are three types of metadata: descriptive, structural, and administrative. Descriptive metadata are used to discover the object. A researcher may use descriptive metadata to limit a search by title and author in an OPAC or other database. Structural metadata define the object's internal organization and are needed for display and navigation of that object. For instance, structural metadata may contain information about the number of pages an object contains and what order they should be viewed in. Administrative metadata contain the management information needed to keep the object over time and to identify artifacts that might have been introduced during its production and management. For example, administrative metadata indicate when the object was digitized, at what resolution, and who can access it.

The project testbed proposes to use existing descriptive metadata standards, such as MARC records and the Dublin Core, as well as standards that incorporate both descriptive and structural metadata, such as the Encoded Archival Description (EAD), to help the user locate a particular digital object. This report proposes defining new standards for the structural and administrative metadata needed to view and manage digital objects.

At a higher level, the report proposes a Digital Library Service Model in which services are based on tools that work with the digital objects from distributed repositories. This approach 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 (for example “Get the next page of this digital diary”). Unlike other models, the Digital Library Service Model includes methods as part of the object.

The report also identifies several archival digital object classes that are being examined as part of the MoA II project, including photographs, photograph albums, diaries, journals, letterpress books, ledgers, and correspondence. One of the objectives for the testbed is to develop the tools that display and navigate these MoA II objects, some of which have complex internal organization. Therefore, another goal of this report is to identify the structural metadata elements that are needed to support display and navigation and ensure that they are included as part of the digital objects. Finally, this report begins to examine the methods (program code) that could be included with each class of object.

After the library and archival communities have reviewed this report, MoA II participants will 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 stimulate discussion of standards for digital library objects and best practices for digitizing archival materials. This discussion must include the project participants, DLF members, and representatives of the wider community. In addition, the project will contribute to the DLF Architecture Committee’s ongoing discussion of distributed system architectures for digital libraries. The MoA II testbed will give the library and archival communities a tool they can use to test, evaluate, and refine digital library object definitions and digitization practices. It is expected that these discussions will move the archival and library communities closer to a consensus on standards and best practices in these areas.

PART I:

PROJECT BACKGROUND

In 1998, the DLF, working with staff of the University of California (UC) at Berkeley, developed a grant proposal that requested support to create a testbed for its Making of America II project. The objective of the testbed was to move the DLF members and the wider library and archival communities closer to the realization of a national digital library by addressing several issues that are critical to this goal.

UC Berkeley submitted the proposal to the National Endowment for the Humanities (NEH), which awarded funding**.** The proposed project team included individuals associated with UC-Berkeley and four other DLF member institutions: Cornell University, the New York Public Library, Pennsylvania State University, and Stanford University.

As described in the proposal, the MoA II testbed is designed to provide a means for the DLF to investigate, refine, and recommend metadata elements and encodings used to discover, display, and navigate digital archival objects. The DLF expects that the MoA II testbed will generate a working system for investigating metadata problems and for discussing, testing, and refining different solutions. The project will give DLF members information that can be used to create the necessary standards or recommendations for best practices for each research area. The project will also be of value to the library and archival communities as a whole because it will advance discussion of the nature of the digital library and move libraries toward a consensus.

The project has three phases: planning, research and production, and dissemination. The planning phase was funded by the DLF. During the research and production phase, which is funded by the NEH and is currently under way, theories developed in the planning phase are being tested. In the dissemination phase, the project will share its tested ideas and practices with the broader community.

Planning Phase (October 1997-May 1998)

Participants in the planning phase decided that the MoA II Testbed Project must engage scholars, archivists, and librarians interested in access to the digital materials represented in the project, as well as metadata and technical experts. The following four activities were recommended:

1. UC Berkeley would work with representatives from Cornell, the New York Public Library, Penn State, and Stanford, and with consultants and selected archivists, to review the collections 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, and ledgers. (The MoA II Steering Committee recommended before the start of the project that books and serial articles be considered outside the scope of this project.)
2. UC Berkeley, working with the same group, would draft a paper that identified the behaviors of each class of digital objects and

the structural and administrative metadata to support those behaviors. In addition, the paper would suggest initial best practices for digitizing the classes of archival objects to be included in the project. Finally, it would include a compilation of existing work in these areas as well as any original contributions the group could provide.

3. The participants in the MoA II Testbed Project and the DLF Architecture Committee would review the draft paper. It would then be revised and distributed to the wider community for review.
4. Technical experts at UC Berkeley would analyze the paper and design a means of encoding the behaviors, metadata, and objects for implementation during the research and production phase of the project.

Research and Production Phase (May 1998-March 2000)

The MoA II testbed would be used to investigate, refine, and enhance the working definitions of administrative and structural metadata, and the important behaviors of archival objects. The testbed project has the following goals, defined during the planning phase:

- to create tools that help the library community understand how digital archival objects are discovered, displaye, and navigated;
- to understand how these tools use metadata and what value the metadata provide and at what cost; and
- to give the DLF a set of metadata practices that can be reviewed and recommended to the wider community.

Dissemination Phase (Summer 2000)

When the research and production phase has ended, the MoA II Testbed Project will seek funding for an invitational seminar at which project results will be reviewed. Participants will include digital library experts, archivists and special collections librarians, scholars, computer scientists, museum professionals, and others who have participated in developing the EAD protocols, are engaged in similar work, or have appropriate expertise. At the end of this phase, project results will be disseminated, practices established will be refined, as necessary, and an agenda for further community review will be formulated.

PART II:

THE MOA II DIGITAL LIBRARY SERVICE MODEL

Overview

The digital library service model developed for the MoA II Testbed Project has three layers: services, tools, and digital library objects (fig. 1). In this model, services are provided through tools that discover, display, navigate, and manipulate digital objects from distributed repositories.

This report also proposes a digital object model that fits within the service model. The object model defines digital objects, which are the foundation of the service model, as an encapsulation of content, metadata, and methods.

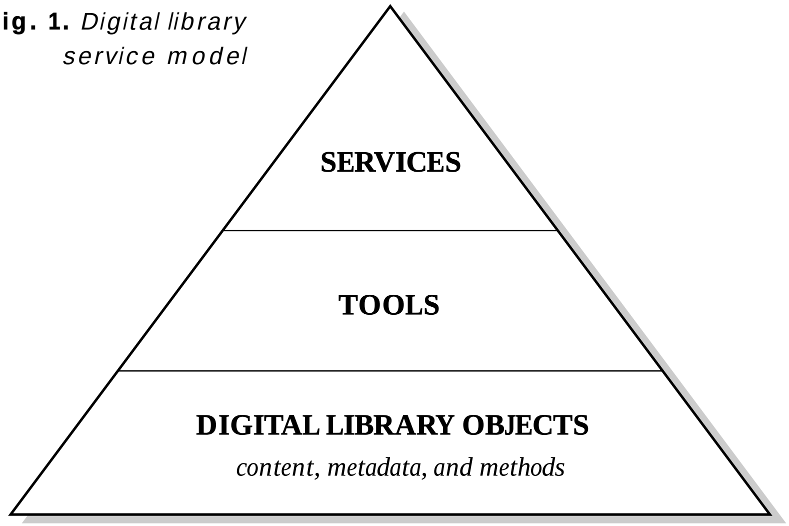
Each of the layers in the model may be described as follows.

Services Layer

This layer describes the services to be provided for a specific group of users. Because the MoA II Testbed Project relates to scholars' use of archival materials, 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 services layer contains a suite of tools to support the needs of a particular group of users. 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.

Fig. 1. *Digital library service model*



Tools Layer

This layer contains the tools that serve the user. The MoA II tools consist of the following:

- an online catalog for the discovery and display of the USMARC collection-level records;
- a standard generalized markup language (SGML)-compliant database that will be used to search, display, and navigate the EAD-compliant electronic finding aids; and
- tools to display and navigate the MoA II-compliant digital archival objects. (Objects are MoA II-compliant when they can be delivered using the proposed encoding standards described later in this paper.)

Any tool is actually a suite of behaviors, or actions. With a digital diary, for example, such behaviors could include actions such as “Turn to the next page,” “Go to the previous page,” “Jump to Chapter 3,” or “Translate this page into French.”

Digital Library Objects Layer

This layer contains the actual digital objects that populate distributed network repositories. Objects of the same class share encoding standards that encapsulate (that is, include) their content, metadata, and methods. Separate classes of digital objects could be defined for books, continuous-tone photographs, diaries, and other objects.

A Model for Digital Library Objects

Digital library objects form the foundation of the digital library service model. It is now possible to 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 (for example, diaries, journals, photographs, and correspondence). Each object in a given class has content that is a digital representation of a particular item. The content can be digitized page images, ASCII text, numeric data sets, and other formats. The following are examples of three classes of archival objects and their content format:

- a photograph made up of a single digitized tagged image file format (TIFF) image
- a photo album made up of 30 photograph objects
- a diary made up of 200 digitized TIFF page images and textual transcriptions

The object model starts by defining classes of archival objects in a system under which 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, metadata are considered as separate from content. Metadata are data that in some manner describe the content. The DLF systems architecture committee has identified three types of metadata:

1. Descriptive metadata are used in the discovery and identification of an object. Examples include MARC and Dublin Core records.
2. Structural metadata are used to display and navigate a particular object for a user. They include information on the internal organization of an object.[1] For example, a given diary has three volumes. Volume I has two sections: dated entries and accounts. The dated entries section has 200 entries; entry 20 is dated August 4, 1890, and starts on page 50 of Volume I.
3. Administrative metadata represent the management information for the object, including the date it was created, its content file format, and rights information.

Metadata can now be added to the model. Any class of archival object encapsulates both content and metadata, where the metadata are used to discover, display, navigate, manipulate, and learn more about a particular object's management information.

The distinction among the three 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 the 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

Several concepts used in this paper, including method*s*, originate from object-oriented design (OOD).

Object-Oriented Design as Part of the Object Model

The popularity of OOD is evident in the widespread use of related programming languages such as C++ and Java. Some of the reasons for this popularity also make OOD an attractive addition to the digital library service model. In particular, OOD actually models users' behaviors, making it easier to more accurately translate their needs

[1] Structural metadata 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 it. At the other extreme, a diary could exist with only enough structural metadata to turn the pages.

into system applications. This advantage will be discussed in more detail later.

Object-oriented design has another important advantage. In OOD, a digital object conceptually encapsulates both content and methods. *Methods* are program code segments that allow an object to perform services for tools. 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).*

The most important advantage of making methods part of the object may be that these basic program segments do not then have to be reinvented by every developer.[2] Instead, the developer can have the tool ask the object's existing method to perform the needed work. This makes the development of new tools faster and easier. Since tools directly support the end user in this model, their development should be encouraged.

Defining the Difference between Behaviors and Methods

One great advantage of the object-oriented design approach is that it models users' behavior with methods. There is a clear distinction between user-level *behaviors* and *methods*. The word *behaviors* relates to 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 into French." The word *methods* refers to how system designers describe what tools can do for a user.

One important reason for distinguishing between behaviors and methods is to establish a process that will enable libraries to engage their users in a dialogue about 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 examples of user-level behaviors might be relevant an to item in the digital library class "diary":

- "Show me the organization of this diary." (It may have three volumes, each of which includes sections 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."

[2] This digital object model is only conceptual. Complete objects made up of metadata, data, and methods do not sit in a repository waiting for use. Instead, they are created as needed. That is, the parts of the objects (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 such as object-oriented databases. Relational databases, for example, could be used for the persistent storage.

- “Show me the fourth journal entry.”
- “Show me the first entry for August 1890.”
- “Show me more entries on the same topic.”
- “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 case, these user-level behaviors would have to be mapped into a series of methods that perform 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 (1) fetch the table of contents and (2) 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

Methods now become part of 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 be present.[3]

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 object’s 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 (data) into a digital library object.[4] The metadata and content must be encoded in a standard manner for objects in a given class. This encoding is required so that the methods defined for each class can work across all objects in that class.

[3] The methods that are part of an object tend to be those that are most used across sets of tools. Tools themselves will have methods and therefore will need access to the metadata and content of the objects. Project staff 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] While the content and metadata must be encoded in a standard manner, they do not necessarily have to be stored together nor do the three different types of metadata need to reside together because objects only come into existence as needed. Therefore, the object can be assembled virtually from persistent storage when required.

Summary

This report proposes a digital library service model for the MoA II Testbed Project in which services are based on tools that work with the digital objects from distributed repositories. This model recommends that libraries begin by defining the services they need to provide for each audience they support. Next, they must define the tools needed to implement these services. This process should include the identification of user-level behaviors for the tools, that is, what the tools do as required by the users. This report 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 (for example, diary or photograph), and the content of each object may be text, digitized page images, photographs, or another format. The object also contains metadata of three types: (1) descriptive metadata used to discover the object; (2) structural metadata that define the object's internal organization and are needed for display and navigation of that object; and (3) administrative metadata that contain management information (such as the date the object was digitized, at what resolution, and who can access it). The digital object definition borrows from the popular OOD model and includes methods as part of the object. Methods are program code segments that allow the object to perform services for tools.

PART III:

IMPLEMENTING THE MOA II SERVICE MODEL

Selecting Digital Archival Classes

Project staff selected a group of object types and classes from the materials suggested by institutions participating in the MoA II project. These archival object types are the core of those examined in this project. The number of items selected was limited to facilitate completion of the testbed within the project time frame.

The object types and classes selected were as follows:

1. **Continuous-tone photographs**: Single archival object. The photograph may have a caption or other textual information recorded on its face or on verso. Continuous-tone photographs are interesting for this project because they exist in abundance in many of the collections and because they enable a close look at the collection of administrative metadata for use in object behaviors. The most basic of objects, the continuous-tone photograph provides a solid platform upon which to base the project.
2. **Photograph albums**: Bound manuscript object containing a collection of continuous-tone photographs. The photograph album may contain captions that are separate from the photographs or other items such as newspaper clippings. Photograph albums are a logical extension of continuous-tone photographs, since they contain photographs that are ordered in a structured manner and that raise both structural and administrative metadata issues.
3. **Diaries, journals, and letterpress books**: Bound manuscript objects, usually arranged chronologically and with date notations. May have additional structure (for example, an “accounts” section noted in the back). These structured documents have 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 the New York Public Library, full searching and navigation are possible.
4. **Ledgers**: Bound manuscript objects that contain accounting records. They are usually arranged by account although sometimes they are in chronological order. The structure of documents of this sort is different from that of diaries and journals; however, in terms of structure and navigation, they may be considered a variation on a theme rather than a different object type. For these objects, inclusion of more text, while costly, allows for more sophisticated searching and navigation.
5. **Correspondence**: Objects of this class may be simple (a one-page letter) or complex (a long letter with an envelope and enclosures). Investigating correspondence allows the project to examine these sometimes-complicated documents and the structural metadata relationships between the subdocuments (letter to envelope, for example).

These types of materials have been selected for the testbed because the participating institutions hold large quantities of them or because they offer the project important challenges in terms of the behaviors needed to view, navigate, or manipulate them. The complex structure of photograph albums, for example, requires that users be able to see individual photographs, a photograph with its caption, photographs and captions in the context of pages, and pages in the context of an entire album. With diaries and journals, users may want to see individual entries or to jump from one entry to another or from an index to an entry. Diaries and journals also raise the issue of individual page scans that do not correspond with the logical structure of the document. For example, entries frequently end in the middle of a page and a new entry begins on that same page. In addition, when different levels of metadata are available, these materials allow for display and navigation experiments. For example, a minimal digital diary might consist of a series of page images with only a base set of behaviors that can be implemented (for example, "turn to the next page"). A richer digital diary may have encoded text transcribed that allows for a variety of tool behaviors for each page image. For example, the tools could display a table of contents for the diary, jump to a particular page or entry, or search for text strings. The structure of letterpress books and ledgers offers the potential for interaction between indexes in a document and its individual entries or parts. The project is also exploring how the structure of these items differs from those of diaries and journals. While ledgers, letterpress books, journals, and diaries are different classes from an archivist's point of view, they may be quite similar from a structural metadata perspective.

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

1. placed into a single base object;
2. created as separate objects (one for each letter) and linked through the creation of a new aggregate collection;
3. created as separate objects inside an embedded collection (folder) object. A collection object has metadata for the collection, followed by the embedded objects, each with its own metadata and content. This approach differs from item 2 in that the objects are embedded rather than linked; or
4. 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 collection object, can be divided into sections by text encoding. For example, a diary can have dated entries identified by the text encoding that can be used for display and navigation. In the same manner, any type of object may also be a compound object through linking to other objects or embedding other objects inside itself.

While the concepts of compound, linked, and embedded objects are not new, the MoA II testbed will give archivists and librarians a tool with which to better evaluate 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. It will also allow for the evaluation of each practice to identify how tools can best use each practice to meet audience needs.

MoA II Testbed Services and Tools

The digital library service model described in Part II of this report is a three-tier model consisting of services, tools, and digital objects. The MoA II testbed is implementing 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 will link to the digital archival objects in that collection.

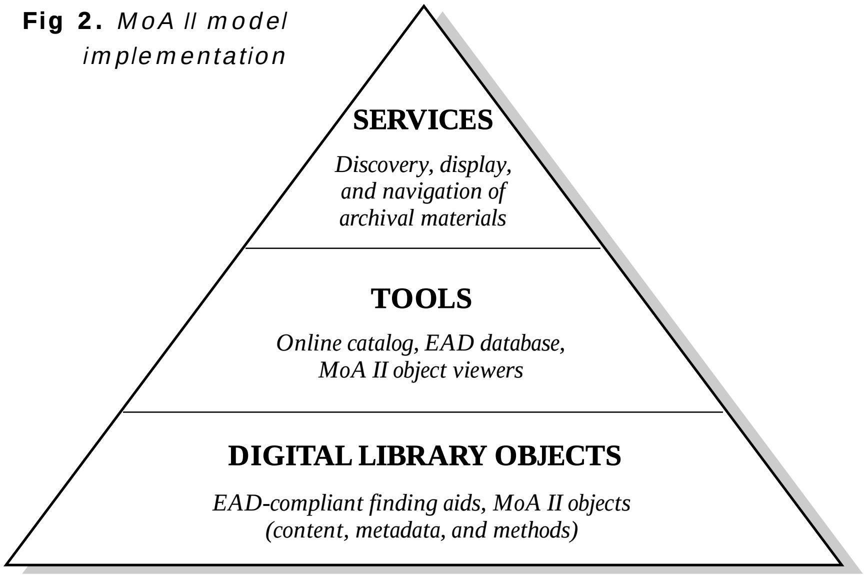
The top tier in the model (see fig. 1) is a services layer composed of suites of tools that focus on supporting particular groups of users (such as scholars, undergraduates, or K–12 students). An archivist, for example, would require different tools for the discovery, display, navigation, and manipulation of digital archival objects than would a fourth-grade student. Initially, the MoA II Testbed Project is focusing on general services for scholars who are using the classes of digital objects selected for the project. Future research projects could include developing service models for novice users or customized services for specialized scholars (such as what is envisioned for the Digital Scriptorium Project).

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

- an online catalog (OCLC’s SiteSearch) used to discover and display the USMARC collection-level records;
- an SGML-compliant database (INSO’s DynaWeb) used to search, display, and navigate the electronic finding aids that are compliant with the EAD; and
- display and navigation tools to be used with MoA II-compliant digitized photographs, photograph albums, diaries, journals, and correspondence. As the project entered the production year, project staff consulted various parties (including archivists, scholars, and librarians) to understand the behaviors required in this tool set.

The MoA II testbed implementation of the digital library service model is shown in figure 2.

Fig 2. *MoA II model implementation*


SERVICES
Discovery, display, and navigation of archival materials
TOOLS
Online catalog, EAD database, MoA II object viewers
DIGITAL LIBRARY OBJECTS
EAD-compliant finding aids, MoA II objects (content, metadata, and methods)

Behaviors and Methods—“What Tools Do”

Part II of this paper introduced the concepts of behaviors and methods in the context of object-oriented design. This section outlines those concepts in detail.

Definitions

The digital library service model defines *behaviors* as the ways in which users describe what tools do for them. Engaging users in a dialogue about behaviors can help identify user needs in terms of the functions performed by tools. System designers can then map these user-level behaviors into *methods*—defined as 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 have the same name as a user-level behavior. The ability to create methods that model a user’s desired behaviors is one of the strengths of object-oriented design.

This definition of methods will be applied to specify the range of activities that should be supported through the metadata described elsewhere in this paper. In an object-oriented model, the methods are embedded in the digital objects; digital objects reveal methods to tools interacting with them. Many repositories in the MoA II environment will not deploy object-oriented models; in this case, the methods will be made available to tools that interact with repositories, rather than with the digital objects themselves. Nevertheless, this model is likely to be 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.

The digital library should support methods that both 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 method might involve providing the tool information about levels and types of organization (such as "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 complex navigation (for example, "Locate the first dated entry in November 1884;" "Find entries where dates are separated by more than 10 days"). Although no object or repository would be required to support the full range of methods, the model proposed here will facilitate the development of increasingly sophisticated tools that can be scaled for use on a growing body of complex archival objects.

Contexts and Constraints

The methods of the digital library reside in tools that may be client-based or server-based, depending on the state of technology. The location of that method (that is, with the client or server) may shift with changes in technology. For example, widespread adoption of an image client that supports progressive transmission of image data might shift image processing from server to client, thereby expediting the processing and reducing the load on the server. However, an interim measure might rely on a server-based compression-decompression process in which the server can generate pan or zoom views at the user's request in real time. This relieves the client of processing responsibility and shifts the work to the server.

Just as the methods may move from client to server and back again, they will also separate into specialized functions or merge into high-level, multifaceted functions. For example, *print* and *display* might be different methods, with one object optimized for screen display and another optimized for printing. In Adobe Acrobat, for instance, *display* and *print* are merged in the same tool. If it is argued that Acrobat handles *display* poorly, a clearer separation of these two methods might be advisable. By separating the methods conceptually, it is possible to assess the applicability of the tool in the service of the method.

High-level methods frequently consist of a series of calls to lower-level methods. For example, because *print* is a behavior that most users require in a tool, most tools will have a high-level method expressed something like "print(a1,a2…aN)." The information inside the parentheses represents arguments that tell the method what object to print, which printer to use, and so on. The *print* method would execute a series of lower-level methods. It may, for example, 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 (spools) the formatted content to that particular printer.

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 primatives. 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 200 by 200 pixels centered on coordinates X by Y."

The following sections define methods that are central to creating the digital library. When possible, the descriptions are sufficiently generic to apply to a variety of objects; in some cases, the methods described are specific to some data types.[5]

Navigation

General Navigation

Navigation consists of a request, a receive, and a display action. 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 navigation. For example, the user who finds a digital scrapbook in a repository may request 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 of a nested list of chapter and section headings. In navigating 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 discernible format; and
3. display that information in a way that is meaningful for the user.

The user expects a series of references, not actual delivery of the objects, which will not be sent until requested.

Navigation also depends on an understanding of relationship primitives—*parent*, *child*, and *sibling*—that 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

[5] *Generic* and *specific* are difficult to define in this context. The discussions that take place in the MoA II process are anticipated to help define and agree on generic methods.

Each major heading may be presented as a link for further expansion, perhaps offering second-level headings to the user (for example, 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. The repository would make a determination that, for example, all first-level and second-level headings fit within the threshold (providing the annual groupings within the dated entries 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. In the scrapbook example cited earlier, the user could be 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, and the programmatic or other purposes of an initiative. Examples include conceptual and structural references (as in the scrapbook example), simple page lists (such as Project Open Book and the University of Michigan Making of America sites), and pages of thumbnail representations of larger format images.

Image Navigation

In contrast to the generic form of navigation just discussed, image navigation uses image-specific information. Systems that display image information need information analogous to that used in geographic references; for example, X and Y coordinates, along with dimensions of the portion of the image to be displayed. Increasingly, tools for image management and manipulation use segmentation to optimize the relatively confined space of a video display as well as other resources in short supply (such as network capacity, memory, and CPU capacity). Some of these technologies are primarily server based, while others shift responsibility to the client. Wavelet compression, for example, allows a repository to store an extremely high-resolution image and generate lower-resolution versions and subsets in real time, at the request of the user or an intermediary. Another approach, implicit in tools or formats such as JPEG Tiled Image Pyramid (JTIP), segments the image into overlapping tiles in a pyramidal structure and allows the user to pan and zoom on a full-resolution image by requesting the next tile or a corresponding tile at a higher resolution. The image-navigation tool receives information about resolutions, resolution ratios, and window sizes to make the navigation possible; the image display tool uses that information to pan, zoom, crop, or otherwise use images.

Display and Print

The display method uses a reference to a known object to deliver an item to a screen-oriented tool, such as a graphical Web browser. In contrast to navigation, where the user tool requests object references, the display tool works from an object identifier it has received from an intermediary or can infer from a query where only one object exists. From this relatively simple operation emerge more complex issues, one of which 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 such as a chapter, again in a format identified by structural metadata. Examples of references that are slightly more complex include requests to display the next or previous sibling of the digital object (next page or previous chapter) or the parent of the digital object (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 (for example, “250 by 250 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 a method similar to the display method; it differs only in its use of printers, plotters, disks, and other output devices. 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 portable document format (PDF) for delivery. The display and print methods are closely intertwined and differ according to the formats available from the repository and user preferences. For example, imagine that a repository of bitonal, 600 dpi page images offers graphics interchange format (GIF) images with interpolated grayscale, Postscript, and PDF. A user without Adobe Acrobat may choose to display the GIF images but print using the Postscript files containing encapsulated images. A user with Acrobat may choose to rely entirely on the PDF image files to 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—most common with art and architectural images (consider the common use of two slide projectors in art historical instruction)—are applicable to both images and text. Support for the comparison of two passages of text or the display of a text alongside an image is also needed. Common applications might include the following:

- 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 (Dante Gabriel Rossetti’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; or

- the display of a number of image objects positioned in a grid of specific dimensions.

Somewhat more complex is the requirement to combine objects. The need to apply layers to, or remove them from, an object is especially important. This problem is considerably easier when a single repository has provided these layers with the base image; however, a more generic method is needed to allow the combination of layers from diverse sources. For example, the allegorical drawings found in nineteenth-century journals such as *Harper's Weekly* often 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 person. A tool supporting the display method might offer that image in any number of different resolutions; it might even display 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 these types of administrative and structural metadata may be too challenging for early portions of the testbed project, the model must be flexible enough to accommodate this information in later iterations.

Repository Search

Repository search methods, more than most other methods, tend to be exhibited by server-side programs rather than by client-side tools.[6] At least portions of these methods could eventually migrate to the user's desktop, but considerable standardization must first take place. In a repository search, an intermediary collects information about the user's query and the characteristics of the available collections and begins to process results (for example, in a sorted list by object or by collection). This intermediary has distinct discovery and retrieval functions.

To support discovery within and among repositories, each collection must participate in a conversation with the client or intermediary. This conversation constitutes the method associated with discovery. The repository's discovery method must include the ability to understand the search parameters of the repository, including gathering information on searchable fields, on the sorts of operators that can be applied, and other constraints. These mechanisms have been specified in protocols such as Z39.50, but it is important to explore

[6] The MoA II project will rely primarily on a union catalog to effect discovery. A union catalog obviates problems associated with inter-repository searches—how one characterizes the search to the various systems and brings 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, while maintaining important elements of institutional autonomy. Repository search, and especially the means to support inter-repository search, is discussed here not because it must be explored in MoA II but because it is an important method to keep in mind when conceptualizing the digital library.

more flexible mechanisms such as the search interface language (SIL) specified in Nigel Kerr's *Personal Collections and Cross-Collection Technical White Paper* (1997).

To support the retrieval of results within and among repositories, the conversation must 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 be found in Nigel Kerr's white paper.

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), which provides 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 that are more likely to be from the same paper stock because of color or texture. CLUT information can also be used to measure subtle variations such as shape and patterns; this would enable the user to detect, for example, hidden features such as characters obscured by palimpsest erasures. Methods using the CLUT could support these types of analyses.

Bookmarks, Annotations, and Links

As the digital library grows in maturity and capability, the array of interactions with objects will become more complex. Users want intraobject bookmarks, annotations, and more sophisticated linking methods, none of which is outside the capabilities of readily available desktop technology. Nevertheless, the ability to support these methods is hampered by unreliable or incomplete metadata, the absence of generalized notions of user authentication and authorization, and a lack of support from repositories. Most important, libraries and archives lack the tools to exploit methods in these areas. Many of these methods are explored in detail in the research applications developed by Robert Wilensky and his team as they pursue notions of "multivalent documents" (Phelps and Wilensky 1996). Moreover, emerging standards such as the extensible markup language (XML) and extensible linking language (XLL) will help articulate 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 through metadata that document a full range of digital object features.

MoA II Metadata

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

Descriptive Metadata

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

In the discovery process, users will search 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. Of course, it will also be possible to search the finding aids directly to discover archival library objects.

Structural Metadata

The terminology of the digital library is evolving rapidly. Consequently, there is considerable variation in how the library community uses certain terms, one of which is structural metadata. (For more information on the definition of structural metadata, readers are referred to Structural Metadata Notes in the Appendix.) For the purposes of this paper, the term *structural metadata* is defined as those metadata that are relevant to the presentation of a digital object to the user. Structural metadata describe the object in terms of navigation and use. The user navigates an object to explore the relationship of a subobject to other subobjects. Use refers to the format or formats of the objects available for use rather than formats stored.

Current thinking divides digital library metadata into three categories (descriptive, administrative, and structural) or two categories (descriptive and structural, with structural metadata subsuming administrative metadata). This report separates administrative and structural metadata. More important than this separation, however, are the categories proposed for inclusion in the MoA II architecture.

Although the categories defined here are presented in SGML, the data in a repository will not necessarily be stored in an encoded form or in a table. This document does not advocate a particular method for storing data; various approaches will be necessary in different institutions, and several approaches may even exist within the same institution. For example, descriptive metadata may be stored in US-MARC, portions of structural and administrative data in relational tables, and other portions in SGML. These examples are intended to illustrate the type of data presented in interactions between repositories and intermediaries. The authors assume that these metadata will be extracted from metadata management systems in interactions with intermediaries such as tools. Further, default and inherited values will be expressed explicitly at the level of the subobject, even if implicitly associated with the subobject through the metadata management system.

Structural Metadata Elements and Features Tables

Tables 1 and 2 recommend the full set of possible structural metadata elements that an individual collection may find useful. Some repositories will use only the minimum set of required elements. Others will also use other elements that can be derived in an automated fashion. Still others will use elements that are easy to capture or derive. The tables include both minimal and maximal values, identify required and repeatable fields, and identify whether field values may be inherited or supplied manually. Some elements can fulfill both administrative and structural functions.

Some elements are relevant to raw data (such as page scans) which do not require extensive examination of the data structure. Other elements are relevant to "seared" data (such as chapter divisions and headings), which require 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), which require serious examination of structure.[7]

Table 1 describes structural metadata defining the object, and table 2 describes structural metadata defining the digital subobjects (for example, the individual digital pages). Thus, the structural information for a digital object is divided into information referring to the constituent objects that cohere into a whole (such as a description of the extent of a digital book) and information specific to the individual parts (such as page or image references).[8] The distinction between object and subobject metadata is in some ways artificial. For example, a tool might assemble information related to the constituent parts of a photograph album by querying each of the constituent subobjects rather than by querying a specially designed digital object. However, certain economies, such as stating ownership only once in the object rather than with each subobject, prevail when storing information. This model strives to balance the specification of elements accordingly.

[7] The University of Michigan (UM) uses the terms *raw, seared,* and *cooked* to describe levels of processing for the Making of America I materials (www.umdl.umich.edu/moa/). For a discussion of MoA I processing at UM, see www.dlib.org/dlib/july97/america/07shaw.html and various sections in http://www.umdl.umich.edu/moa/about.html.

[8] For its organization and for many of the elements, this model owes a great deal to the *Structural Metadata Dictionary for LC Repository Digital Objects*. Located at http://lcweb.loc.gov:8081/ndlint/repository/structmeta.html.

Administrative Metadata

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

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

Some metadata elements are 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 the format of a particular file.

Administrative metadata are 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 file type, creation date, source of original, methods or personnel that might have introduced artifacts into the image, and location where different parts of this digital object (or related objects) reside. Eventually, administrative metadata may help support the long-term management of objects; for example, the metadata contained within the objects will allow for automation of migration from an older file format to a newer one, for refreshment, and for regular backup.

In the past, certain administrative metadata (such as file formats) resided in file headers, while others resided in accompanying databases. In the future, all administrative metadata may reside within the file header. Such a system would be ineffective, however, until there are community standards that specify where they would go within the header, how to express them, and so forth. Work is already under way to develop those standards. In April 1999, NISO, DLF, and RLG convened a meeting to discuss technical metadata elements for images. The results of the meeting are available on the Web (Bearman 1999). For the purposes of this paper, the necessary administrative metadata fields are defined without regard to a particular syntax of where they will actually reside. For the purposes of the MoA II project, administrative metadata will be delivered external to the image file header.

This section primarily concerns administrative metadata for master files. In the future, however, repositories are likely to see master files that are themselves derivatives of previous files. To make the administrative metadata that the authors identify as compatible as possible with future developments, some information dealing with derivative files (or other instantiations of a work) is included. In this way, the project will lay the groundwork for future research projects to identify and trace the provenance of a particular digital work.

Administrative Metadata Elements and Features Tables

Tables 3–6 recommend the full set of administrative metadata elements that an individual collection may find useful. Some repositories will use only the minimum set of required elements; others will also use elements that can be derived in an automated fashion. Still others will choose to use elements that are easy to capture or derive. The tables include both minimal and maximal values and specify which are allowed to repeat. Some elements can fulfill both administrative and structural functions.

Although the number of metadata fields seems daunting, a high proportion may be the same for all the images scanned during a particular session. For example, metadata about the scanning device, light source, or date are likely to be the same for an entire session. 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 field for each digital image; instead, they can be handled either through inheritance or by batch-loading various metadata fields. This report attempts to identify best practices for metadata development. Individual repositories will follow these practices to the extent that they can afford.

Tables 3–6 describe elements needed for the creation of a digital master image; identification of the digital image and tools for viewing or using it; linking the parts of a digital object or its instantiations; providing context; and ownership, rights, and reproduction information. Tables 3 and 4 show the type of data that uniquely identify a particular representation of a work. For future derivative images, these could be iteratively nested to represent the provenance of a work.

Encoding: Best Practices

Encoding Archival Object Content and Finding Aids

Many MoA II materials require 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 text via optical character recognition (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. There has been substantial work and community effort in both areas, and efforts are under way to organize discussions around the use of the available guidelines.

Project participants should use the EAD for the encoding of finding aids.[9] While 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 under way in the DLF about interinstitutional searching of EAD-encoded collections and the resulting scrutiny of local practice. Using locally defined needs to drive the application of EAD will help clarify the range of needs for interinstitutional applications.

Text encoding efforts in MoA II will be well supported by the SGML articulated in the Text Encoding Initiative Guidelines (TEI).[10] The TEI support a range of document types and methods, including the transcription of primary sources and damaged documents. More important, the TEI Guidelines and the associated DTDs (document-type descriptors) offer support for encoding the range of possible structures in MoA II documents, whether or not transcriptions are included. The TEI, like the EAD, offers considerable flexibility in the ways documents can be encoded.

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. XML-compliant versions of both the TEI and the EAD DTDs are likely to be available soon. One editor of the TEI guidelines has been centrally involved in writing the XML specifications, and the TEI editors have declared their intention to create XML-compliant versions of the widely used TEI DTDs.

Encoding to Encapsulate Metadata and Content Inside the Archival Object

After this report has been circulated and discussed, the project team will have gathered enough information to define an encoding scheme for the archival objects that will populate the MoA II testbed. A draft of the MoA II XML DTD has been completed and readers are referred to view both the DTD and documentation available at http://sunsite.Berkeley.EDU/MOA2/ (see the section on MoA II Tools). This XML DTD will define the transfer syntax used for MoA II objects. Selecting an XML to encode the object does not mean that any repository must use that encoding for internal object storage. However, it does give the DLF and the larger community an opportunity to discuss and evaluate XML as transfer syntax.

[9] Information about the EAD, including guidelines for the application of the EAD and DTDs, is available at http://lcweb.loc.gov/ead/.

[10] Information about the TEI can be found at the TEI Consortium home page http://www-tei-c.org/. A searchable version of the *TEI Guidelines* can be found at the TEI Consortium home page, and via the Humanities Text Initiative pages at http://www.hti.umich.edu/docs/TEI/.

Table 1

STRUCTURAL METADATA ELEMENTS
object level

Element	Comments	Required	Repeatable	Source
Unique Identifier Reference	A unique identifier must be presented with each digital resource. 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	Could be automatically generated
Content Type	This describes the types of content available (i.e., not necessarily the 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), and (2) the format available for each. Objects will be available in several formats. For example: Page: TIFF, GIF, JPEG, PDF Image: JFIF, JTIP, etc. Text: HTML, XML Computationally, a tool or service *should* 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 4 wide and 324 high." It may not appear as a *table*, per se, but as options available for selection by a user.	Yes	Yes	Manually supplied in data capture
Extent	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: **Tool:** *How are you organized?* **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. **Tool:** *Okay, let's go for descriptive organization. Give me all of the section headings because there are fewer than 100.*	Yes	Yes	Manually supplied in data capture

Table 2

STRUCTURAL METADATA ELEMENTS
sub-object level

Feature	Comments	Required	Repeatable	Source
Data File Size	This indicates 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 were 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 *use* 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 this book). This carries a numeric value only and must be specified.	Yes	No	Manually supplied in 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, 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 *arbitrarily* 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 a 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 *or* automatically generated

Table 3

ADMINISTRATIVE METADATA 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	This identifies the material from which the digital file was created—the item on hand, even if it is a reformatted version. For example, 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	This indicates the actual physical dimension scanned if cropped, and is needed for appropriate facsimile output.	Yes	No	
Source Characteristics	Film type/ASA/manufacturer, print type; tightly bound volume	This is relevant if an intermediary stage is actually removed from "original." It may add descriptive details of source that could affect scanning quality.	No	No	
Source ID	A local catalog unique ID for a book; an accession number for a special collections item	This indicates the source of the source image (recursively).	Yes	No	
Scanning Date	Any usual time stamp: yyyy/mm/dd or yyyy/mm	Year/month specificity is needed to assist in later evaluation of technology at the time of digital master creation.	Yes	No	Automatically generated
ICC Scanner Profile		This describes the color artifacts introduced by the scanning device. It is necessary to map the images into standard color space and to adjust for display and printing devices.	No	No	Generated once per session
Light Source	Example: 3400K Tungsten, infrared, Osram Delux L fluorescent	Specific settings should be given for the scan (f-stop, electronic shutter speed, filtering, illumination level) since they may be necessary in later evaluation of color capture. They may be specific to each image or by inheritance to collections of images via a separate descriptive file (with anomalies indicated per image as needed).	No	No	Batch generated
Resolution	600 dpi; 400 dpi interpolated to 600 dpi; 1536 x 1024	This indicates the settings on the input scanning device. Cameras usually measure them in dimensions, other devices in dpi. Note where a device does its own interpolation.	No	No	Automatically generated

Table 4

ADMINISTRATIVE METADATA ELEMENTS

for identifying and viewing the digital image files (master or derivative)[1]

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

[1] These metadata are 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.

Table 5

ADMINISTRATIVE METADATA ELEMENTS

for linking the parts of a digital object or its instantiations, providing context (overlaps with structural)

Element	Examples/Definitions	Comments	Required	Repeatable	Source
Overall View Image	The image file representing the overall view (used to find the location of a detail within the overall image); may not apply if details are not used	If this image file represents a detail or part of another image, the parent file should be indicated here.	No	No	Manually supplied in data capture
Sequence Number	Example: 4 of 5; 2B of [Primary Image ID]	This indicates the relative position of a particular image in an image file chain that begins with the file named in the PRIMARY IMAGE tag.[2]	No	No	Automatically generated
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)	This indicates the total number of image files in a given sequence; the number may not be known at the time of capture but should be 'calculated' before using the master or beginning derivative production.	No	No	Automatically generated
Version (from MESL Data Dictionary)	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.	No	No	Manually supplied in data capture
Version Date	DEFINITION: mm/dd/yy.	This shows the date the version referred to in VERSION was created; it could be somewhat redundant if this data is used in the VERSION field	No	No	Automatically generated

[2] 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, e.g., wide view to narrow view, left to right, top to bottom, or no particular order.

Table 6

Administrative Metadata Elements

for linking the parts of a digital object or its instantiations, providing context (overlaps with structural), and 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.[3]	No	Yes	Likely to be inherited for an entire collection
Owner Number	Example: [varies widely]	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 to be inherited for an entire collection
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 to be inherited for an entire collection
Credit Line (from MESL data Dictionary)	DEFINITION: The text required to be displayed whenever the image/data appears. EXAMPLE: Copyright Berkeley Art Museum, 1978. All rights reserved.		No	No	Likely to be inherited for an entire collection
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 to be inherited for an entire collection
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 to be inherited for an entire collection
License Term	DEFINITION: specifies the duration of any licensing arrangement covering this image		No	No	Likely to be inherited for an entire collection
License Begin Date	DEFINITION: start date of any licensing agreement covering this image expressed as mm/dd/yy		No	No	Likely to be inherited for an entire collection
License End Date	DEFINITION: end date of any licensing agreement covering this image expressed as mm/dd/yy		No	No	Likely to be inherited for an entire collection

[3] This 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.

APPENDIX:

STRUCTURAL METADATA NOTES

1. "Structural metadata is metadata that describes the types, versions, relationships and other characteristics of digital materials" (Arms, Blanchi, and Overly 1997).

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, and Overly 1997).

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, and Overly 1997).

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 1997).

5. "Looking at the larger picture, there are three type of "metadata" which have been identified by the National Digital Library Project of the U.S. 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 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 1997).

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" (World Wide Web Working Group, Cornell University Library, 1996).

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)" (Cornell University, Distillation of the Working Group Recommendations, 1996).

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. A 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 1997).

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 1997).

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 1996).

REFERENCES

Organization of Information for Digital Objects

Arms, William, Christophe Blanchi, and Edward Overly. 1997. An Architecture for Information in Digital Libraries. *D-Lib Magazine* (February). Available at http://www.dlib.org/dlib/february97/cnri/02arms1.html.

Lagoze, Carl, Clifford A. Lynch, and Ron Daniel Jr. 1996. The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata. Available at http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Body/ncstrl.cornell/TR96-1593/html.

Payette, Sandra, Christophe Blanchi, Carl Lagoze, and Edward Overly. 1999. Interoperability for Digital Objects and Repositories: The Cornell/CNRI Experiments. *D-Lib Magazine* (May). Available at http://www.dlib.org/dlib/may99/payette/05payette.html.

Metadata

Bearman, David. 1999. NISO/CLIR/RLG Technical Metadata for Images Workshop, April 18-19, 1999. Available at http://www.niso.org/imagerpt.html.

Gartner, Richard. 1997. SGML as Metadata: Theory and Practice in the Digital Library (session abstract). Available at http://users.ox.ac.uk/~drh97/Papers/Gartner.html.

Kerr, Nigel. 1997. Personal Collections and Cross-Collection Technical White Paper. Available at http://dns.hti.umich.edu/~nigelk/work/pccc.html.

Morgenstern, Matthew. 1997. A Framework for Extensible Metadata Registries. From Metadata Registries Workshop, University of California at Berkeley, July 1997. Available at http://dri.cornell.edu/Public/morgenstern/registry.htm.

Perkins, Alan. 1997. Information Warehousing: A Strategic Approach to Data Warehouse Development. Visible Systems Corporation White Paper Series. Available at ftp://www.infoeng.com/whitepapers/DW.pdf.

Structural Metadata Dictionary for Library of Congress Repository Digital Objects:

- Data Attributes: http://lcweb.loc.gov:8081/ndlint/repository/attribs.html.
- Structural Metadata List: http://lcweb.loc.gov:8081/ndlint/repository/attlist.html.
- Data Definitions: http://lcweb.loc.gov:8081/ndlint/repository/attdefs.html.
- Examples of using this model:

— for a photo collection: http://lcweb.loc.gov:8081/ndlint/repository/photo-samp.html.

— for a collection of scanned page images: http://lcweb.loc.gov:8081/ndlint/repository/timag-samp.html.

— for a collection of scanned page images and SGML-encoded, machine-readable text: http://lcweb.loc.gov:8081/ndlint/repository/sgml-samp.html.

World Wide Web Working Group, Cornell University Library. 1996. *Metadata Working Group Report to Senior Management.* Available at http://www.library.cornell.edu/DLWG/MWGReporA.htm. See also Distillation of the Working Group Recommendations, Appendix 1 in Report of the November 11, 1996, Professional Staff Meeting on the Cornell Digital Library. Cornell University Library, November 1996. Available at www.library.cornell.edu/DLWG/DLMtg.html.

Scanning And Image Capture

Besser, Howard. 1996. Procedures and Practices for Scanning. Canadian Heritage Information Network (CHIN). Available at http://sunsite.Berkeley.edu/Imaging/Databases/Scanning.

Besser, Howard, and Jennifer Trant. 1995. Introduction to Imaging: Issues in Constructing an Image Database. J. Paul Getty Trust, Getty Art History Information Project. Available at http://www.getty.edu/gri/standard/introimages/index.html.

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

Fleischhauer, Carl. 1998 (July 13). “Digital Formats for Content Reproductions.” National Digital Library Program, Library of Congress. Available at http://lcweb2.loc.gov/ammem/formats.html.

Fleischhauer, Carl. 1996. Digital Historical Collections: Types, Elements, and Construction. Washington, D.C.: National Digital Library Program, Library of Congress. Available at http://lcweb2.loc.gov/ammem/elements.html.

Seaman, David. Image Scanning: A Basic Helpsheet. Electronic Text Center, Alderman Library, University of Virginia. Available at http://etext.lib.virginia.edu/helpsheets/scanimage.html.

International Color Consortium: http://www.color.org.

International Organization for Standardization. 1998. Graphic Technology—Prepress Digital Data Exchange—Tag Image File Format for Image Technology (TIFF/IT). Technical Committee 130, ISO/FDIS 12639. Geneva: International Organization for Standardization.

Kenney, Anne R. 1996. *Digital Imaging for Libraries and Archives*. Ithaca, N.Y.: Cornell University Library.

Phelps, Thomas A., and Robert Wilensky. 1996. Toward Active, Extensible, Networked Documents: Multivalent Architecture and Applications. *In Proceedings of the First ACM International Conference on Digital Libraries*, 100-8. Bethesda, Md., March 20-23, 1996.

Picture Elements, Inc. 1995. *Guidelines for Electronic Preservation of Visual Materials*, revisions 1.1, 2. Report to the Library of Congress, Preservation Directorate, March.

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

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

Reilly, James M., and Franziska S. Frey. 1996. 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 the Image Permanence Institute, May. Available at http://lcweb2.loc.gov/ammem/ipirpt.html.

Technical Recommendation for Digital Imaging Projects. 1997 (April 2). Image Quality Working Group of ArchivesCom, a joint Libraries/AcIS Committee. Available at http://www.columbia.edu/acis/dl/imagespec.html.

"Text Scanning: A Basic Helpsheet." Electronic Text Center, Alderman Library, University of Virginia. Available at http://etext.lib.virginia.edu/helpsheets/scantext.html.