Implementing Encoded Archival Description: An Overview of Administrative and Technical Considerations

MICHAEL FOX

Abstract: The implementation of Encoded Archival Description involves the same programmatic and administrative concerns encountered when undertaking any new initiative. To be successful, EAD must fit within the institution's budget, priorities, and strategic vision; benefits must be carefully weighed against costs. Administrators need to understand the implications of initial and ongoing staff costs, outsourcing possibilities, the role of consortial initiatives, training requirements, workflow management, hardware and software requirements and costs, and technical support needs as they select from among several options for creating and publishing EAD-encoded finding aids.

About the author: *Michael Fox is Head of Processing in the Minnesota Historical Society's Division of Library and Archives. A member of the EAD development team and SAA's EAD Working Group, he has been both an early implementer and an instructor for numerous workshops on EAD.*

ENCODED ARCHIVAL DESCRIPTION (EAD) is a tool for making archival finding aids such as inventories and registers available for search and display in an electronic format. The process of implementing EAD is not fundamentally different from the launching of any new program or initiative. Many of the programmatic and administrative issues presented here will be familiar from other contexts, with the added complications inherent in the adoption of any new technology.

Is EAD right for your archives? This article identifies the ingredients necessary for successful implementation of this technology as a way of helping archivists and managers answer that question. It first addresses programmatic and administrative concerns, followed by an overview of technical issues, including a description of several methods for "publishing" electronic finding aids. The structure of this review may also serve as a framework for considering the six case studies that will follow in the fall 1997 issue of the *American Archivist*.

Programmatic Issues

Implementing EAD is a more complex undertaking than simply buying computer software and sitting down to mark up finding aids.[1] To ensure success, an archives must first address the fundamental issues raised by the following questions: Does EAD meet our institutional mission, goals, and strategies? What resources will be required? How will we manage and carry out the work?

Mission, Goals, and Strategies

It is axiomatic that an archival institution should undertake no new initiative without first carefully considering whether it conforms to the organization's larger mission, goals, and strategies. EAD implementation is surely no exception; the use of EAD makes sense only if the actual benefits correspond to institutional objectives. What then, does EAD offer? Your assessment might focus on the following characteristics and merits. EAD is:

- A descriptive standard for finding aids that enables multiple uses of the information they contain, their interchange, and their long-term accessibility;
- A communication format for finding aids that enables archives to deliver them electronically to distant users;
- A technology that is standards based, computer platform independent, and employs powerful tools for the searching, retrieval, display, and navigation of finding aids.

Answers to some pointed questions about your clientele, and the ways in which your inventories are now used and their current condition, will help frame your response.[2] Candor is required. Is the digital delivery of both metadata and collection materials themselves an important goal for your institution? How might the delivery of searchable, electronic inventories fit into that objective? Are off-site users a target audience? Who uses your inventories? How often and in what ways are they used (to identify box numbers, to make copies for researchers)? How many finding aids do you have? Are you confident

[1]This article uses interchangeably the expressions finding aids, inventories, and registers.

[2]The author wishes to acknowledge the "EAD Implementation Checklist" developed by Helena Zinkham and others at the Library of Congress as a source of many useful recommendations incorporated into this article. This checklist is available within the FAQ section of Anne Gilliland-Swetland, "Encoded Archival Description Document Type Definition (DTD) Application Guidelines," unpublished draft disseminated electronically, December 1996. Available at <http:/scriptorium.lib.duke.edu/findaids/ead/guidelines/index.html>.

about their quality and completeness? If they are less than optimal, would you be willing to share them in their present condition? If not, how much revision would be required? In what physical format are they? In light of your answers, are the long-term benefits of EAD worth the effort involved?

For many institutions, adherence to standards is a key strategic goal when deploying new technology. It is seen as insurance that protects one's investment in technology by making it possible to take advantage of a broader and more diversified marketplace, while also enhancing one's ability to migrate data to future systems. For decision makers such as library and archives directors who have adopted this strategy, the standards-based, community development aspects of EAD will provide a convincing rationale for its adoption.

Priorities and Resources

Simply assessing the benefits is not enough; even highly desirable projects must be weighed against other useful activities as one sets priorities and allocates limited resources. Such planning decisions are subject to an array of local variables that are beyond the scope of this article, but the relative costs of a project typically are one significant factor. The major costs for EAD will be the purchase of the technology (hardware and software) and staffing.

Staffing

It probably will come as no surprise that personnel costs will be by far the largest expense, though hard figures are difficult to come by for several reasons. The pool of archives with EAD experience is relatively small. Personnel costs are very institution-specific (for example, colleges and universities may have access to a talented but relatively inexpensive student labor pool). EAD work absorbed by current staff may appear to be "free."

To help in calculating staff costs, I will describe the principal tasks that may be necessary for implementing EAD. The full recitation of these activities may seem overwhelming—even discouraging at first and perhaps beyond the reach of many archives. However, this overview lays out many options, only a subset of which will actually have to be acted upon. The experience of the Special Collections Department of the library at the University of Vermont, as told by Elizabeth Dow in her paper appearing in the fall 1997 issue of the *American Archivist*, illustrates how much a small archives can do.

Prior planning is vital if costly missteps are to be avoided. Many issues must be resolved before the first finding aid is mounted on a computer. All of the case studies that will be in the fall 1997 issue of the *American Archivist* speak forcefully about the need to plan. A thoughtful review of the status quo will be a good beginning. What is the role of finding aids in your institution's reference and access system? Do the catalog and inventories work together in an integrated search process? Where does their content overlap? Are they created in the most efficient manner, given their interrelationship? Dennis Meissner's article in the fall 1997 issue of the *American Archivist* describes how the Minnesota Historical Society spent considerable time rethinking the structure and presentation of its finding aids before beginning markup. This included asking fundamental questions about the purpose of inventories, their relation to collection-level descriptions in MARC, and their physical appearance. These were evaluated in terms of the informational content of the finding aids, how users in the reading room perceived them, and how remote access might affect their use.

Staff must evaluate hardware and software requirements and then select, acquire, and install the tools. As we shall see later, there are multiple options for creating EAD-encoded finding aids and making them electronically available to users. Archives will have to assemble their own systems from a mixture of components as there are no real turn-key systems available at this time. It will take staff time to evaluate these choices and reach decisions. As always with technology, this process is complicated by rapid evolution in the computer marketplace, which tends to cloud choices. If EAD implementation occurs as part of a multiple-repository effort or other consortial project, one must factor in the overhead attendant on resolving these issues in a collaborative context.

Once implementation is underway, the project must be appropriately managed. The work of purchasing equipment, contracting for services, negotiating with partners, and hiring and supervising staff is not a trivial matter. Ongoing operations require personnel to mark up, proof, and test the finding aids; load and manage files on computer servers; and handle any associated image files. Quality control will be important for any work that is contracted out. If the finding aids are to be electronically linked to the on-line catalog, the relevant MARC records must be edited to add the necessary pointers. As will be discussed later, one might also wish to supply an HTML version of each EAD finding aid for users with older Web browsers that do not support SGML or XML. If so, a process for converting the EAD files into HTML must be developed and implemented, as Nicole Bouché describes in her paper on Yale's implementation of EAD in the fall 1997 issue of the *American Archivist*. Many early implementers of EAD are offering explanatory materials on their websites that describe what finding aids are, how they are used, and methods for access.[3] This text must be prepared, encoded, and loaded.

Technology-based programs are always dynamic, subject to continuous evolution and revision. Depending on the options chosen, as described later, the level of technical expertise required to deliver finding aids electronically will range from modest to substantial. Archivists are resourceful people, and many of the early implementers of EAD have undertaken their projects with existing staff who have acquired new computer skills. Repositories that have access to in-house technical support staff, perhaps from a parent organization, may be able to tap into those resources. Purchase of support services is always a possibility, especially where it involves standard computer activities such as setting up desktop computers and servers or installing and configuring common tools such as Web servers. SGML database search engines may be less familiar to contract personnel, and so such services may be more difficult to locate or more expensive to acquire.

Training

Staff education and training needs are easy to overlook. Besides acquiring a working knowledge of the structure and content of the EAD standard, those who will work with EAD must master specific software packages for creating, converting, and editing finding aids. The Yale, Harvard, and Library of Congress implementation case studies appearing in the fall 1997 issue of the *American Archivist* describe a process of collaborative self-education. The technical computer skills required will be described in greater detail in the technology section of this paper. As usual, training options range from reading the software manuals to enrolling in formal classes or workshops.

[3]For examples, see the Yale EAD site at <http://webtext.library.yale.edu>.

Outsourcing

Contracting for services is a popular approach to special projects and may be appropriate for some aspects of EAD implementation. The choice between performing work in-house and contracting it out is usually one of trading time for money. Doing the work oneself usually involves less out-of-pocket expense but consumes precious staff resources. In some institutions, it may be easier to obtain funds such as grants, gifts, or extraordinary budget allocations for "special projects" such as implementing EAD than it is to hire additional regular staff. Some tasks, such as text markup, database installation and administration, and Web server maintenance are obvious choices for outsourcing.

Some level of involvement by regular staff is inevitable, however. Planning and operational oversight are difficult, if not impossible, to contract out. Moreover, the contracting process itself generates administrative overhead. Certain staff skills are required: a knowledge of the issues involved; the ability to articulate institutional objectives and convert them into clearly measurable vendor deliverables; familiarity with contract negotiations; and an understanding of the dynamics of contract supervision, especially where quality control is an issue. A successful vendor-customer relationship requires both parties to have a clear and detailed agreement on their objectives and requirements. This would be particularly important, for example, in outsourcing EAD encoding services, given the wide range of choices that EAD provides for marking up finding aids. The options exercised in this area will directly and perhaps significantly affect the time required to encode an inventory, and with that the cost of conversion.

Cooperative Ventures

Early implementers of EAD, institutions large and small, have found direction, support, and funding in consortial undertakings, joint projects, and other shared ventures. These include cross-campus cooperation at Harvard and Yale, as well as multi-institution projects such as the American Heritage Virtual Archive Project (Berkeley, Duke, Stanford, and Virginia) and the University of California EAD Project (nine campuses). While the view of these repositories may be that they are large and well staffed, Leslie Morris' article in the fall 1997 issue of the *American Archivist* reveals that they are often just a loose federation of smaller operations (one to three persons) whose staff are as thinly spread across multiple responsibilities as in any small archives. The ways in which they have been able to work together to plan for, educate themselves about, and implement EAD suggests a cooperative model that others may beneficially replicate. This approach will be particularly useful for planning, procurement, and technical support.

Workflow

One hallmark of success for any new project is its ability to incorporate new activities into existing operations without significantly adding to daily workloads. While the planning and managerial activities previously described are significant, they largely represent an initial investment of time that need not be repeated, or at least continued at the same level of intensity, in the future. Ongoing operations, chiefly the encoding of inventories and the maintenance of the computer infrastructure, are where the real increase in effort is likely to occur. The most efficient implementation, therefore, will be one in which those activities can be closely integrated into, or simply replace, existing tasks on a one-for-one basis. Most archives today create printed inventories with a word processor. If one can continue to create finding aids in the same way and convert them afterwards into EAD (as Yale and the Minnesota Historical Society do), or substitute an SGML editor for the word

processor (as Harvard does), the impact on workflow may be minimal. Indeed, the net effect of such changes may actually be greater efficiency and less work. Meissner reports on the use of standard word processing templates at the Minnesota Historical Society, where their use appears initially to have reduced the time needed for keying container lists and the subsequent cleanup of data entry errors.

Conversion of existing finding aids raises another group of prioritization and workflow issues. Encoding newly processed collections will be relatively straightforward and "clean;" new work, new methods. Marking up preexisting finding aids is more complex, raising at least three sets of issues: priorities, techniques, and editorial and stylistic revision. Realizing that the conversion of older inventories may go on for an extended period, how does one set priorities? Technically, what is the best method of converting existing finding aids, both those already in word processing format and those available only in paper form? How much editing to match current institutional practices is feasible? These issues will be familiar to repositories that have implemented MARC cataloging and undertaken the associated retrospective conversion of their catalogs.

Whether one converts all or only some existing inventories, it is necessary to select those to be done first. Early implementers of EAD have taken several approaches to this matter. Selection criteria may focus on materials relating to a particular topic, activity, or group; for example, the Minnesota Historical Society has begun to convert collections documenting environmental and natural resources issues. One might emphasize the "significance" of collections for research. Focusing on the potential for enhancing access is another possibility. The ability to perform text searches of the contents of an inventory will yield greater rewards for some finding aids than for others. For example, searching a container list that consists chiefly of an enumeration of box numbers, volume titles and span dates, as one often encounters in certain types of government and organizational records, may not significantly enhance subject access. The availability of an existing electronic version of the text could be another determining factor, as David Seaman's article in the fall 1997 issue of the *American Archivist* reports was the case at the University of Virginia.

The level of tagging to be used is also an important workflow consideration, one discussed extensively in the Library of Congress, Harvard and Yale case studies. Protocols developed range from minimal content designation to experiments with richer markup, such as tagging each instance of personal and geographic names wherever they appear in the inventory. These decisions affect current encoding costs and, at least potentially, future retrievability. The Library of Congress staff describe the numerous hypertext links embedded within their finding aids, a task requiring additional data entry and testing.

Technical Issues

The programmatic and planning concerns addressed thus far share many traits in common with other institutional initiatives. In addition, EAD implementation requires consideration of the full range of technical issues that surround the introduction of any new computer technology. The staffing implications of the procurement, installation, and management of systems have already been described. This section focuses on the operational side of technology: how one creates encoded documents, disseminates them to users, maintains the files, and otherwise manages the technical system that supports these activities.

The technical aspects of EAD that lie ahead (for the implementer and for the reader of this article) are both complex and ambiguous. They are complex because there are so

many options, because the technical details may be difficult to visualize without actually seeing them (especially for those not versed in EAD), and because the terminology of converters, styles, templates, and macros may be unfamiliar to the reader. They are ambiguous because, lacking a full understanding of how electronic finding aids actually may be used, we cannot fully anticipate all the consequences of the decisions that must be made today.

Authoring EAD Documents

First, one must create encoded finding aids; multiple approaches are available. Each will be described, including a list of its strengths and weakness and its suitability for creating new inventories (for collections that are being processed for the first time or where typed inventories are being rekeyed into electronic form), as well as the method's potential for converting existing electronic files. The authoring choices fall into five categories: native SGML authoring packages, SGML-aware text processors, word processor add-ons, other text processing tools, and databases.[4]

Native SGML Authoring and Editing Software. Native SGML authoring and editing software, such as SoftQuad's Author/Editor, ArborText's ADEPT Editor, or Incontext2 from InContext Systems, may be used to key inventories as an alternative to using word processing programs such as WordPerfect or Microsoft Word. While specific features vary somewhat, all SGML authoring packages include a Macintosh or Windows graphical interface for WYSIWYG[5] keying and editing of inventories. Typically, data entry begins with a user-created template on the screen that contains commonly used EAD elements, much the same way a cataloger uses a workform that displays a set of the most often used MARC tags. The text of the inventory is keyed into the appropriate tags. At each point in the inventory, the software displays a list of the currently valid elements from which the typist selects the correct tag; the software thus assists the user during the encoding process by enforcing compliance with the structural syntax of EAD and the inclusion of mandatory attributes and elements. These products are also parsers; that is, they check the finished document for conformance with the DTD. This assures that a valid SGML "instance" (i.e., a finding aid document) is produced, one that can be successfully displayed, indexed, shared, and retrieved. The software prevents the typist from making encoding errors during data entry that would require subsequent adjustments (an ounce of prevention is a useful thing!). Other important features include the ability to create templates for standardized content and layout, including the insertion of "boilerplate" text, and the use of keyboard macros to avoid repetitive key strokes and to speed data entry. Documents are produced directly in SGML for permanent storage.

As with any software, there is a learning curve in mastering authoring software. The experience of archivists who have used them in EAD workshops suggests, however, that they are no more challenging than word processors and are perhaps less confusing to learn, as they have fewer bells and whistles. User-defined data templates may be employed to make the work easier for data entry operators who are not trained archivists, but some basic understanding of the hierarchical structure of container lists will be required.

Most institutions will want to create nicely formatted print copies of their inventories in addition to electronic versions. Authoring packages vary in their capabilities for pro-

[4]This taxonomy is the author's; other writers may choose to categorize these tools in other ways.
[5]"What you see is what you get."

ducing such printing details as running headers and footers, pagination, graphics and fonts. SoftQuad, for example, offers an interface to the desktop publishing software Quark as a means of producing more sophisticated print output from its Author/Editor software. ADEPT Editor has a companion product, ADEPT Publisher, for generating print output. Incontext2 has its own style language to generate well-formatted print copy.

SGML authoring software also may be used for the conversion of existing inventories that are already in an electronic format or that can be scanned and passed through optical character recognition (OCR) software to produce readable text. Two approaches are possible. One may create a template containing empty EAD tags and cut-and-paste the existing electronic text into the appropriate tags. The other choice would be to ''wrap'' the preexisting text, in situ, with the proper EAD tags. While the use of keyboard macros can expedite this process, this approach requires a strong knowledge of the EAD tag set to avoid creating parsing errors in the process. Converting the container list portion of an inventory will always be the most time-consuming and relatively complex part of the work, whichever solution is employed.

SGML-aware Text Processors. SGML-aware text processors such as Framemaker+SGML from Adobe and recent versions of WordPerfect offer another alternative for authoring EAD documents. These differ from SGML authoring packages in that they are a standard desktop publisher and word processor, respectively, that have the ability to work directly with an SGML DTD such as EAD.

In many respects, WordPerfect functions very much like authoring software. Prior to creating inventories, one loads a copy of the DTD into the software so that it can intelligently manipulate files. A dialog box with a list of valid elements appears in a window, and the typist selects the appropriate tags as data entry proceeds. But WordPerfect goes one step further by applying styles to format the document simultaneously for printing. Styles are a convention employed in many word processors that define how certain parts of a document appear on a printed page or the screen. For example, the style assigned to an individual paragraph can govern the size and type of the font employed, line and paragraph spacing, tab setting, indenting, and other display characteristics. In WordPerfect the user may create a ''layout file'' that defines an appropriate display style for each EAD element. In this manner, one creates an EAD SGML file and, at the same time, a properly formatted WordPerfect document.

This method has three distinct advantages. There are the synergy and time savings of an integrated solution. The use of familiar tools may reduce start-up costs for software and training, enhance staff acceptance of new processes, and integrate easily into existing workflow. These packages also have the built-in ability to generate nicely printed output. The downside is that they require more work for initial set up. Someone on staff will have to learn features of the software such as styles, templates, and other conventions. As no standard templates exist for inventories at this time, each institution will need to create its own initial mappings. While most word processors have the capability to create templates and styles, an informal (and admittedly less than scientific) poll of participants in EAD workshops reveals that few archives have made use of these features in the past.

Word Processor Add-ons. The third method for producing EAD documents is a variation on the former approach, one that might be characterized as the use of post-authoring converters. Microsoft's SGML Author for Word and Microstar's Near and Far Author enable one to convert documents produced using the Windows versions of the popular Microsoft Word word processor.

It may be easiest to imagine this scenario as the reverse of the previous. With WordPerfect, one encodes a document in EAD and associates the SGML elements with particular display styles. Using Word add-ons, one creates a document using the styles feature of Word and then maps the styles to EAD elements. A Word template is created that defines a separate style for each part of the inventory that corresponds to a particular EAD tag. For example, one might create a style called "C01Title." That style would then be assigned to the text of the finding aid that would later be encoded as <c01><did><unittitle>. The text converter is then programmed, through an interactive editor, to associate particular Word styles with corresponding EAD tags. For instance, one would instruct the software that the text in style C01Title should be encoded as the element string <c01><did><unittitle>, and so forth throughout the document. As the finding aid is typed in Word, one assigns the proper style to each section of text. When done, one simply saves the file as SGML and the converter program outputs a correctly tagged EAD instance using an association file to map Word styles to the appropriate EAD elements. The Microsoft Word converters and the SGML-aware text processors also can convert existing SGML files into their native Word, WordPerfect, or Framemaker format respectively.

This class of software also can be used for the conversion of existing finding aids. One imports an existing document, applies the appropriate styles to each section of text, and runs the conversion program. The Minnesota Historical Society experience suggests that while this process requires manual intervention and some sense of the EAD hierarchical structure, it requires only a basic knowledge of the tag set. Where lengthy container lists contain simple folder listings, whole blocks of text may be highlighted and converted to an appropriate style in one step.

Other Text Processing Tools. The fourth authoring scenario involves the use of other text processing tools and techniques. The simplest way (and least expensive in terms of software cost) to mark up documents is to use a basic text editor like Notepad (which comes with every copy of Windows), to type EAD text, beginning with "<ead><eadheader><eadid>MHS75-0005798</eadid></eadheader>. . .", etc. This is a perfectly possible but complex, exacting, and time-consuming solution; one that requires a detailed knowledge of EAD to create parsable documents. A number of freeware parsers are available for the technically adventurous who wish to use this approach.

Other possibilities exist as well. For example, University of California library staff have created a tool using "perl" (a text manipulation computer programming language) that generates finding aids through the use of a fill-in-the-boxes screen template. A program written in perl does the rest of the work, generating an EAD document.[6] One might call this "markup for dummies," as it requires little knowledge of EAD structures, but writing perl "scripts" is a fairly sophisticated programming exercise that will be beyond the skill or training level of most archivists. The perl programming language also has proven useful for converting text already in electronic form into EAD. Other text manipulation programs may be used as well. At least one institution has achieved the same effect using the macro language of Microsoft Word to convert word processing documents into EAD. The Minnesota Historical Society uses this tool to translate SGML documents into HTML format. Bouché describes how Yale relies on the use of the macro language in WordPerfect and Edix/Wordix for conversion of existing text. The 8.0 version of WordPerfect also includes

[6]The templates generated for the University of California EAD Project may be viewed at <http://sunsite.berkeley.edu/FindingAids/uc-ead/templates/>.

extensions to its macro language to make it more SGML-aware. There are also commercial text manipulation programs such as DynaTag from Inso Corporation and OmniMark from OmniMark Technologies that can perform this same function. They are powerful and efficient, particularly because they are SGML-aware and understanding the basic concepts of wrapping text and nesting elements that are fundamental to SGML structures such as EAD.

These text manipulation solutions typically utilize "visual" clues such as the formatting of text, punctuation, and the use and location of tabs, paragraph markers, and line breaks to make "educated guesses" about the content and structure of a document in order to convert it into SGML. The more consistently one's inventories have been physically laid out on the page, the more successful these techniques will be across a corpus of existing finding aids, as described in Bouché's report of the Beinecke Library's conversion experience.

Databases. The last option is the use of databases to generate EAD files, creating documents in a proprietary database format and converting them to EAD when exported into a text file. Two commercially developed products fall into this category: Gencat from Eloquent Systems and Internet Archivist from Interface Electronics. There are substantial differences between the two products. Gencat is a proprietary database package that can be used for the creation, storage, searching, and delivery of descriptive information about archives at a variety of levels, from collection- or fond-level descriptions to container lists. As such, Gencat is a full-fledged authoring and publication system. As an additional feature, it can export this data in different formats, including MARC and EAD, though its ability to do the latter has not yet been fully demonstrated. Internet Archivist, now in the final stages of development, is strictly an authoring tool that stores the EAD finding aid in its own database structure for convenience but exports the file as an SGML document. From the user's point of view, the software functions just like the native SGML authoring or text processing tools that feature a fill-in-the-boxes interface. Its strength lies in the fact that it is the closest thing to a turnkey authoring package available, as it was designed to work specifically with EAD; as such, it will be simple to install and operate. The developer has announced its intention to produce additional modules that will provide data import and conversion services and the capability to search across finding aids. The ability of these two packages to export data in a standard format such as EAD will certainly offset some of the concerns about the long-term viability of storing one's data in a proprietary database structure.

Some further words of caution about databases are in order. They are powerful tools for data management, but they do have limitations. Archivists who like to develop their own databases are cautioned against assuming that they will be able to incorporate the hierarchical structure of EAD easily and cheaply into an application built with off-the-shelf database management software such FileMaker Pro or Microsoft Access. This is not to say that it cannot be done, but the effort would be significant.

Publishing Inventories

"Publishing," or the electronic delivery of finding aids to users, is the second technical aspect of EAD implementation. There currently are at least three ways, described here as scenarios, of accomplishing this; they are not mutually exclusive.

Publication of EAD-encoded finding aids does not necessarily require use of the World Wide Web. After all, SGML products preceded the appearance of the Web, and other distribution methods such as CD-ROM are possible. However, the ubiquitous pres-

ence of the Web and the widespread use of browser software for the dissemination of information suggest that we should focus on these tools. One important forthcoming development that will have considerable impact on making EAD accessible over the Internet is Extensible Markup Language (XML). A simplified "dialect" of SGML, XML can be thought of as SGML-Lite for the Web. XML will have an impact on the authoring and electronic publishing of EAD finding aids in two ways. First, additional authoring tools will certainly emerge. Second, and more importantly, Web browsers such as Netscape's Navigator and Microsoft's Internet Explorer will be able to read EAD files in SGML/XML directly without requiring that they first be "dumbed down" to HTML. Even so, there will still be a transitional period when some Web users have XML-aware browsers, while others have not yet upgraded to the newer software.

Since these browsers currently cannot read full SGML files, archivists who wish to use the Internet must adopt other solutions in the interim. Either the SGML files must be converted into the simpler HTML tags that current browsers can read, or the user must load and configure a software helper application such as Panorama Pro or MultiDoc Pro. Several options for the former are described in the following scenarios.

Scenario One: Access Through A Website. Many archives have access to an Internet website, either their own or that of a parent organization. In this scenario, the Internet-searching patron locates a page on the archives' website that lists those collections for which an electronic finding aid is available, clicking on a collection name to display a copy of the inventory in the user's browser. The archives may choose to provide an SGML/XML or an HTML version, or to offer the reader a choice of formats as Yale University, the Library of Congress, and the University of Vermont do.[7] Alternatives to a simple alphabetical listing of collections are possible, including groupings by time period, locality, or subject focus. Links to finding aids might also be embedded in an on-line bibliography or topical collection guide.

This scenario is the simplest and least expensive option to implement, but it also offers the lowest level of searching access. Users must either browse the finding aids or know from other sources which collections are appropriate to their needs; searching is limited to the text of only one individual finding aid at a time.

Requirements:
Encoded finding aids.

- Location on a website containing a listing of collections.
- Storage space for the finding aid files on a hard drive accessible by the Web server that serves that website.
- A process for converting EAD files to HTML, if that display option is offered.
- Stylesheet and navigator files to support the display of SGML/XML documents.

Scenario Two: Access from a Web-based On-line Catalog. Many on-line library and archives catalogs now have public interfaces that use a Web browser to access their MARC-based holdings. These are generally available to all Internet users. MARC field 856 permits a cataloger to embed a link in the catalog record, in the form of a uniform resource locator (URL) address that points to another electronic document such as an EAD

[7]For examples, see their websites at <http://webtext.library.yale.edu>, <http://lcweb.loc.gov/rr/ead/eadhome.html>, and <http://sageunix.uvm.edu>.

finding aid.[8] In this scenario, the patron searches the MARC descriptions of archival collections in the on-line catalog using the searching capabilities of the catalog software. When the entry for a relevant collection is displayed, the reference to the electronic version of the finding aid appears as a highlighted browser hyperlink; clicking on that link loads the finding aid into the user's browser. The archives may chose to supply the finding aid in either SGML/XML or HTML format.

This scenario continues the two-step discovery process patrons have used for years in many repositories. First, relevant collections are identified by a search of the catalog, then an inventory that contains greater detail than the catalog record is consulted to narrow the search and select appropriate files. In this scenario, these two steps now occur on-line, possibly far from one's reading room.

For institutions with Web-accessible catalogs, this option offers substantial benefits. Prior investment in a familiar tool, the MARC catalog, is leveraged to provide public access to finding aids. The summary descriptive information in the MARC records offers access by topics, provenance, and other criteria and serves as a useful search filter. Some developers are concerned that the results of a search across the full text of multiple finding aids would produce overwhelming results—a little like trying to take a sip from a fire hydrant. This viewpoint is reinforced for many by their experiences with Web search engines. Providing access to electronic finding aids via an on-line catalog minimizes this concern. For those with a Web-based catalog, costs will be modest.

Requirements:

- Encoded finding aids.
- Web-based on-line catalog with MARC records for archival holdings.
- Ability to update catalog records to provide appropriate hyperlinks.
- Storage space for the finding aid files on a hard drive accessible by the Web server that serves that website.
- A process for converting EAD files to HTML, if that display option is offered.
- Stylesheet and navigator files to support the display of SGML/XML documents.

Scenario Three: Access Through Finding Aid Databases on the Internet. There are several software "search engines" that index and distribute text documents, such as SGML finding aids, via the Internet. Users can search them with a standard Web browser. A single query of such a "database" searches the full text of multiple finding aids simultaneously and returns to the user a list of relevant collections. This activity mirrors what happens when one queries a library catalog that then displays a list of all the titles that match the search. Such a finding aid "database" may contain all the inventories of a single archives or function as a "union database" for multiple institutions. The files need not reside on a single server. At least one product simply stores a central index to multiple finding aids that are themselves physically located on servers at other institutions and which are retrieved only in response to a particular search request. This eliminates the need for each archives to continually transfer new or updated finding aids to the central server.

[8]For examples of links from on-line catalogs, search records in these collections at <http://webpac.library.yale.edu> or <http://scriptorium.lib.duke.edu>.

This approach affords detailed searching of all the richness of the finding aids themselves, across collections and institutions, to a depth of detail never before possible.[9] Four products of this type are in current use: PAT and LiveLink from OpenText, DynaWeb from Inso Corporation, and Site Search from OCLC. All vary in significant details such as ease of configuration and hardware requirements, as well as in the details of "publication" such as the use of stylesheets and the dynamic conversion of files to HTML. DynaWeb, for example, converts SGML files to HTML "on the fly" so that users do not require any special software beyond the customary Web browser. These products are not inexpensive; expect hardware and software costs to begin at $20,000, though Inso does make "grants" of free software (DynaWeb) to "educational" institutions. Dow reports how the University of Vermont received one such grant in her article in the fall 1997 issue of the *American Archivist.*

Other options are feasible, though it is unclear which archives, if any, are currently using them. Text search engines that are not "SGML-aware" might be used, including products such as Folio and Star, that could import and translate SGML files into their proprietary formats, and provide a Web-based search interface to the files, although they will undoubtedly lose the hierarchical structure of EAD in the conversion. If the archives converts the SGML files into HTML format, a locally mounted Web-based search and retrieval engine such as AltaVista could provide indexing and display.

Various public interfaces for searching are under development at individual archives and at the Research Libraries Group as institutions test the various ways in which users might wish to query finding aid databases. For better or worse, user interaction with online catalogs is better known, if less than optimal, for archives patrons. We are only beginning to understand how researchers will react to these new searching opportunities.

Requirements:

- Search engine software and hardware.
- Encoded finding aids.
- Stylesheets for the display of SGML/XML files.
- Web server that can access the search engine.
- Conversion routines for translating SGML files into HTML.

Possible Future Scenario. The Z39.50 standard facilitates computer-to-computer communication. Its most widespread application is facilitating the searching of multiple databases, such as library and archives catalogs, without having to use the particular search syntax of each. Work may begin shortly on the development of a Z39.50 "profile" for EAD that would add finding aids to the list of data types that could be handled directly by Z39.50 databases and browsers.

Conclusions

The issues raised in this article may seem daunting, even insurmountable. The need to describe many options may leave a misimpression that there are thousands of decisions to be made, which is not the case. There are three large decisions to be reached: Shall we do it? How will we create electronic finding aids? How will we distribute them? Choices

[9]For examples, see the following websites: <http://scriptorium.lib.duke.edu:8000/dweb_help/dweb_searching.html> and <http://hul.harvard.edu/dfap>.

will become clearer as more institutions begin implementation, as consortia of archives develop support systems, and as the tools and underlying SGML/XML applications move more prominently into the Internet mainstream.

The adoption of EAD is in an initial phase. Perhaps it is still most appropriate for the technically adventurous archives, large or small. Some institutions may wish to defer full implementation until issues surrounding the technical infrastructure for creating and publishing EAD files are more clearly defined, or until professional support systems such as cooperative projects are in place to provide direction and succor. But is not too soon for any archives to begin to understand EAD and the implications for standardized practices that it suggests, to begin to evaluate local practices in advance of later adoption of EAD, and certainly not too soon to contribute to the community discussion of what our users need in electronic information systems. EAD implementation is a function of the entire archival community, as well as a programmatic decision of individual archives.