Preservation Video File Format Issues and Considerations

What Have We Encountered as We Drafted MXF AS-AP?

Federal Agencies Audio-Visual Digitization Working Group
Carl Fleischhauer, Coordinator, August 15, 2011 (3)

This recaps a series of five messages sent to a listserv related to the MXF Application Specification for Archiving and Preservation (AS-AP), and adds a note on metadata. For background, see http://www.digitizationguidelines.gov/guidelines/MXF_app_spec.html.) The Working Group circulated version 1.d of AS-AP in October 2010 and, as we continued our development during 2011, we encountered some elements for which the resolution is not obvious or that depend on some action by another body. These issues were described in the five messages. Version 1.h of AS-AP posted at our Web site in August 2011more or less "punts" the issues described here. We welcome comments and advice from colleagues.

Table of Contents

Closed captioning and ancillary data

There are number of engineering nuances concerning closed captioning and other ancillary data but--roughly speaking--this comment is about the data found in analog video signals in the vertical blanking interval (VBI) and about the similar data in the digital realm referred to as VANC (for vertical ancillary data) and HANC (horizontal ancillary data). This section of this document does *not* concern vertical interval timecode (VITC), discussed in the next section.

The first-order concern for Working Group members is the reformatting of older videotapes. Nevertheless, we want our MXF application specification for archiving and preservation (AS-AP) to be extensible in the future and, in a modest way, we have started to think ahead to motion picture subtitles and other forms of embedded content. Meanwhile, our expert consultant Oliver Morgan has outlined for us the multiple embedded-data specifications in play in nations beyond North America, including the European Broadcast Union's Subtitling Data Exchange Format (EBU Tech 3264,[1] aka *EBU STL*).

[1] http://tech.ebu.ch/docs/tech/tech3264.pdf

Our discussions of ancillary data brought us to two sets of preservation-related questions: (1) about the data as data and (2) where to put the data in the AS-AP preservation file. Regarding the data itself, we ask:

- Is embedded ancillary data of interest; ought it be retained?
- Are some types of ancillary data more important than others?
- Are there classes of content for which the retention of (some or all) ancillary data is important, and other classes for which retention is not important?

We believe that ancillary data includes information that is needed to properly understand and manage video content objects for the long term. In some cases, the embedded data may be construed to be an essential part of the original item that must be migrated forward in order to create an authentic and complete copy. In some cases, the embedded data contains information (closed captions or subtitles, other descriptive information) that--once ingested and indexed into a search system--will support researchers who seek to discover relevant materials. The embedded data is also likely to contain technical information that will support the management of the item and may also shed light on production-method or provenance topics of interest to researchers.

The second set of questions concern the structure and location of the ancillary data in the MXF file (or perhaps in a "sidecar" file). We have found ourselves looking at multiple options. The first option embraces *current practices* and echoes the wording found in AS-03, the "broadcast delivery" MXF application specification.[2] The AS-03 option applies especially well to US-standard video:

> If present, CEA 608 line 21 (CC and XDS) data shall be carried in a SMPTE ST 334-1:2007-and-ST 334-2:2007-compliant ANC packet within a SMPTE ST 436:2006-compliant VBI/ANC GC Element, using 8 bit encoding. If present, CEA 708B DTV captioning data shall be carried in a SMPTE 334-1:2007-and-ST 334-2:2007-compliant ANC packet within a SMPTE ST 436:2006-compliant VBI/ANC GC Element, using 8 bit encoding. Caption language shall be specified using AMWA AS-04.

The second option pertains to an *emerging practice* that should accommodate both US and European formats. It would employ the structures being standardized in SMPTE-TT Timed Text standards (ST 2052-0:2010, ST 2052-1:2010, and RP 2052-10:2010). Members of this family are being published during 2010 and 2011.

Related to the two previous options--development is not far enough along to see exactly where this initiative will end up--is a European Broadcast Union project to standardize the carriage of EBU STL inside MXF files. At this writing, there is a nineteen-page draft recommendation titled "Transport of Subtitles using MXF in an IT-Based Television Production Environment" that includes this interesting slant on the problem space: "When following the MXF internal approach, six different options are available to store subtitles inside one MXF file." Participants in the EBU project include broadcasters and equipment vendors and they have invited input and comment. We see value in encouraging comment from preservation-minded memory institutions. Oliver Morgan is connected to the STL-MXF effort and has said he will ask if the

[2] http://www.amwa.tv/downloads/specifications/AMWA-AS-03-Delivery-Spec-1_0.pdf (consulted August 9, 2011)

draft document can be shared with our community and will pass along the names of any individuals who would be willing and able to participate.

Meanwhile, as a sidenote, we report that one member of the Working Group has said that his organization will retain embedded metadata "in the raster" after reformatting, in addition to extracting and storing it using whichever of the two options outlined above is finally selected for the published version of AS-AP. This reference to raster-based data pertains to the closed captioning on line 21 of the (former) US analog broadcast standard, and also to data that may be on lines 19 and 20. In addition to closed captioning, broadcast-related video may carry other types of data in these lines, including Nielsen Media Research identification codes and Automated Measurement of Lineups (AMOL) data, which support Nielsen's identification of what program was shown on what channel at a particular time, employed by their rating system.

Finally, we plan to acknowledge the existence of a current practice among broadcasters even as we deprecate it. This is the practice of creating sidecar files that contain VBI and related data. Examples include a pair of ad hoc file types: *vbi* files, readable in some video server systems, and *stl* files, a class of text-plus-timecode-plus-control-character format files. In addition, we have heard references to similar use of *scc* Script Files from the Scenarist application and *srt* files from the SubRip program.

Timecode

A variety of timecode formats may be encountered in the analog and digital video recordings that an organization plans to reformat. These include Vertical Interval Timecode (VITC), Linear Timecode (LTC), Digital Vertical Interval Timecode (DVITC), and other timecode data carried in the Vertical and Horizontal Ancillary data areas in a digital recordings (VANC and HANC). There are also expressions like GCSys timecode (related to the MXF Generic Container) and timecode associated with Captioning Distribution Packets as defined in SMPTE ST 334-1 and -2. For video materials, the code may be "drop frame" to help you stay in sync with a real clock, or "non-drop-frame" when you need a sequence of continuous frame numbers for the whole recording. (This is especially important for U.S. standard television in which the per-second frame rate cannot be expressed as an integer.) Meanwhile (and not our *immediate* concern), there are other forms of timecode data, e.g., keycode or edgecode associated with film or high end digital recordings employed in the production of theatrical motion pictures.

In many cases, a videotape--the original item to be reformatted--will carry a single instance of timecode or, for some ethnographic field recordings, amateur recordings, and the like, there may be no timecode. Meanwhile, some video recordings may include multiple instances of timecode. Timecode may provide a record of the time that has elapsed from the start of the recording (very common), or it may be referenced to a clock, e.g., time-of-day timecode that references "the clock on the wall," actual studio time. Timecode is often continuous through a recording but it may be interrupted or discontinuous. Timecode may also reference other things, e.g., telemetry data in the case of scientific or engineering documentation, and it may include other data in the timecode stream.

Eric Wenocur of Lab Tech Systems in Silver Spring, Maryland, commented on the matter of multiple timecodes:

> Often videotapes have multiple timecodes that do not match simply because one ended up there by accident. A typical example would be a tape where the LTC and VITC don't match because the operator only intended to record LTC and was not careful about how the VITC got recorded. Or the tape had been edited and one of the two timecodes was not regenerated at each edit. . . . There are also cases where the timecode is out of phase, or not locked at all, with the video on the tape. This mostly happened in formats like 3/4-inch when the timecode had to be fed in externally and the generator was not locked to the video source. This can cause problems in editing but might not matter for viewing purposes. . . . [but] in terms of preserving the code, it probably makes sense to get as much as possible into the MXF file.

The Working Group sees the value in retaining or migrating timecode data when reformatting video. Although the need to do this may not apply for all classes of content, carrying timecode into the reformatted copy will often be a part of making a complete and authentic replicate of the original. The potential long-term research value of timecode is evident in the two examples cited at the end of the paragraph at the bottom of the preceding page: time of day and telemetry reference. Such data would help a researcher answer questions like, "When was this recording actually made?" or "Let's correlate *this* recording of the event with some *other* recording or documentation." In addition, timecode may be referenced in descriptive metadata, e.g., the starting and/or ending points for segments. Or the "old" timecode may have been typed into a text transcript that is to be preserved at the same time as the video itself. A scholar may have used the "old" timecode when citing a video segment in an article. Later, a reader may wish to use the timecode reference in order to view the segment after reading the article.

One of the virtues of the MXF format is that it offers a standardized way to include and label multiple instances of timecode. Generally speaking, the production of an MXF file will entail the establishment of what is called *synthetic timecode*, which provides a record of the time that has elapsed from the start of the recording (and could "reuse" or "ditto" an existing elapsed-time stream if a suitable one exists). This main synthetic timecode is referenced in the MXF file's Header Metadata and may also be placed in the file's Material Package.

Our Working Group accepts the value of synthetic timecode but, as the preceding paragraphs suggest, we have also discussed whether and how to migrate pre-existing timecode data from the source recordings to the reformatted copies. Our understanding is that such timecode can be included in timecode tracks within the Essence Containers, labeled for identification as specified in SMPTE ST 377-1:2004.

Interlacing with JPEG 2000 encoding

We want AS-AP to embrace a number of different picture encodings, including JPEG 2000, the topic of this section. JPEG 2000 features a wavelet transform that can be applied in an irreversible manner, resulting in lossy compression, or in a reversible manner, producing lossless compression. Institutions like the Library of Congress are drawn to lossless encoding for moving image content because it reduces file sizes without an adverse impact on quality. James Snyder, the Senior Systems Administrator at the National Audio-Visual Conservation Center in Culpeper, Virginia, uses a compression factor of 2.3:1 as a rule of thumb for the 10-bits-per-sample transfers he prefers. Using that figure, Snyder reports, the storage requirement for

standard definition drops from 121.5 GB/hour uncompressed to 52.83 GB/hour compressed; high definition drops from 668.25 GB/hour to 290 GB/hour. To date, the Library has produced more than thirty thousand losslessly compressed files.

Various members of the moving image community have been developing digital cinema and broadcast profiles for JPEG 2000. The most recent publication is amendment 3 to part one of the ISO/IEC 15444 standard, titled *Profiles for Broadcast Applications* (ISO/IEC 15444-1:2004/Amd 3:2010). Two of the seven profiles in amendment 3 feature the reversible wavelet transform, i.e., lossless compression. Meanwhile, in the MXF context, the relevant standard is SMPTE ST 422:2006, *Material Exchange Format – Mapping JPEG 2000 Codestreams into the MXF Generic Container*. This specification was developed by the digital cinema community to support their use of MXF-wrapped JPEG 2000 picture data (lossy compression) in the Digital Cinema Package.

In the course of our exploration of the intersection of JPEG 2000 and MXF, we have encountered one technical matter that is not fully resolved by the existing standards documents: the handling and labeling of interlaced video. Currently neither the MXF nor JPEG2000 standards documents have specified how interlace fields shall be wrapped in MXF.

Oliver Morgan reports that there are a number of ways that one could encode and label interlaced JPEG 2000 picture data in the MXF setting. The following paragraph offers Morgan's shorthand list of options for frame-wrapped treatment, which is more likely to be used than clip-wrapped in this context. The abbreviation J2C stands for JPEG 2000 *codestream*, as distinct from the JPEG 2000 *files* that are used in still imaging contexts. KLV is an abbreviation for Key-Length-Value (a method for expressing metadata) but in this context KLV is shorthand for the packaged units of data within the file, each accompanied by a KLV unit of metadata.

Here are Morgan's six ways to framewrap interlaced JPEG 2000 picture data in MXF, for US-standard video (30 frames per second/60 interlaced fields per second):

- one field in each J2C, one J2C in each KLV, edit rate = 60000/1001, field dominance indicated in metadata (using Dominance property)
- one field in each J2C, one J2C in each KLV, edit rate = 60000/1001, field dominance not indicated in metadata
- one field in each J2C, one J2C in each KLV, edit rate = 30000/1001, field dominance indicated in metadata (using Dominance property)
- one field in each J2C, one J2C in each KLV, edit rate = 30000/1001, field dominance not indicated in metadata
- one field in each J2C, two J2C in each KLV, edit rate = 30000/1001, linecount property = number per frame
- one field in each J2C, two J2C in each KLV, edit rate = 30000/1001, linecount property = number per field

Morgan noted that there are other options as well, notably a similar list for European 25 frames per second (50 interlaced fields per second). Less plausible are such options as “deinterlaced data, one frame in a J2C”, or "no attempt to deinterlace, one pair of fields in a J2C." Morgan's own working preference has been typed into page 7 of version 1h of the draft specification:

"Interlaced picture data in JPEG 2000 encodings shall be interleaved as field pairs, with the pair wrapped in one KLV unit and with field dominance labeled."

Overall, what is the risk here? When different approaches are adopted by different production systems, and if the labeling is inadequate, files will not interoperate, i.e., a file produced on system A will not play back properly on system B.

What standards body should deliberate this question? One technical expert in the ISO/IEC JPEG 2000 community noted that this is not an issue for picture encoding: "It has to do with how you put the stream in a file," he said, "and that makes it an MXF problem," adding, "it will not be part of our work as we continue to develop broadcast profiles." We agree: there is nothing wrong with the existing MXF or JPEG 2000 standards--what is missing is a specification that maps interlaced JPEG 2000 imagery into MXF. We were pleased to learn of recent SMPTE plans to specify an approach (or approaches) for JPEG 2000 interlaced picture data in a to-be-drafted revision to SMPTE ST 422:2006. Since digital cinema consists of progressively scanned images, the initial drafting of ST 422 did not specify the handling of interlaced picture data.

Operational Patterns

When reformatting videotapes--our bread-and-butter use case right now--the AS-AP files we produce will generally be simple in form, with a single picture essence and a single sound essence. Thus: Operational Pattern OP1a. However, some members of the Working Group see the archiving and preservation file as a useful place to encapsulate items that are associated with the main content element. Examples of associated materials include documents found in the original container along with the videotape, printed matter associated with a sound or moving image publication, transcriptions of oral history recordings, and detailed metadata of special "local" interest. For files with associated essences, the normal MXF categorization would be OP1b. This categorization would also be expected for files containing essences that are members of a related series. Meanwhile, we also foresee segmented essences, e.g., the digital copies of reels within a movie or clips that are cut together into a composite reel. For these, OP-2 patterns would be the norm.

In this mood, we composed this paragraph for our specification-in-progress:

> Baseline Operational Patterns. AS-AP files shall comply with MXF Operational Pattern OP1a (SMPTE 378M), OP1b (SMPTE 391M), or OP2a (SMPTE 392M). AS-AP files shall be labeled as OP1a, OP1b, or OP2a files in the Operational Pattern property of all Partition packs and the Preface Set. AS-AP files shall also include a DMS-AS-AP Descriptive Metadata Set within the MXF Material Package that indicates which specific AS-AP shim applies to the file, as described in the section on DMS tracks below.

For the time being, this seems like a good place to be. But the two expert members of our study committee have reminded us that there is a certain level of volatility here, as sketched below.

How solid are MXF operational patterns as implemented? Oliver Morgan has observed that the implementation of MXF operational patterns has become less rigid during the last few years. In addition, he says, some patterns are widely supported while others are not, with OP-1a the most widely supported (by vendor systems), and OP-Atom a close second. Meanwhile, Snyder

Snyder pointed out that some current implementations that produce OP-1a are not 100 percent reliable, and thus the files created are not 100 percent interoperable. Morgan adds that, in his view, the AS-identifier will be more significant than the OP-identifier.

The claimed simplicity and universality of OP-1a is very attractive. Morgan notes, however, that the limitations of OP-1a swiftly become evident in practice. Referring to the EBU STL subtitling data exchange format,[3] for example, the perfectly reasonable idea of adding STL captions to video would produce a type of file that, in Morgan's words, "is outside the letter of the OP1a law." In some circles, he adds, the increasing importance of including associated files, such as EBU STL or other captions, transcripts, reports, and so on, is encouraging folks to embrace OP-1b or even more capacious operational patterns. The AMWA AS-02 specification (*MXF Versioning*, drafted but not yet published) provides a good example of this trend.

What about DMS-Segmentation? Morgan's reports to our group also called attention to what is called DMS-Segmentation, where some producers employ cut-list metadata for segmented essences. He pointed to relevant wording in AS-03,[4] an MXF application specification published by Advanced Media Workflow Association:

> The Descriptive Metadata Scheme shall be labeled as DMS-Segmentation (UL to be published in the SMPTE Labels Registry). The timeline track shall be constructed of a sequence of DMSegments (or subclasses thereof) or Fillers, following the MXF timing model as described in SMPTE 377-1-2009. SOM is inferred from the start position of each DMSegment, and EOM from SOM plus Duration. (from section 5.4.3, pp. 9-10)

We will consider adding such wording to our application specification, with appropriate caution. Morgan pointed out that DMS-Segmentation still stands in need of more thorough definition, i.e., that more guidance is needed about when to use DMS-Segmentation and when to use OP2 or OP3. He characterized two schools of thought. First, some were looking at simple OPs like OP-1a as capable of carrying essences that are acted upon by what amount to EDLs. They want an MXF file with an essence for which, in some contexts, playout would consist of the presentation of a segment only, using EDLs associated with the DMS. In contrast, the other school of thought prefers to use more complex MXF operational patterns to represent/express segmented material and segmented playout.

Manifests

As reported above, although our bread-and-butter use case is a file with a single picture essence and a single sound essence, we also want AS-AP to embrace additional associated essences (described above) and ancillary metadata, e.g., detailed metadata of "local" interest. We also want the specification to cover segmented essences within a single file, e.g., the digital copies of reels within a movie.

The MXF standard (and its AAF parent) supports the assembly of multi-essence files, providing structure via such elements as operational patterns and *packages*: material, file, and source. The metadata about the structure is provided in the *MXF metadata header set*, along with descriptive

[3] http://tech.ebu.ch/docs/tech/tech3264.pdf
[4] http://www.amwa.tv/downloads/specifications/AMWA-AS-03-Delivery-Spec-1_0.pdf

and administrative metadata. One of the fundamental working concepts guiding our development of AS-AP is that it is a good thing--for preservation's sake--to bind or encapsulate "everything" in a single file. Therefore, we are glad that the structural metadata will be in the file. *Sustainability* is supported by what's *in* the package.

Meanwhile, our expert consultant Oliver Morgan tells us that some members of the broadcast and video-production community seek an additional and/or more facile expression of structural metadata. Working examples often use XML encoding and support the functionality of production and playout activities. Similar desires have also turned up in other digital-content quarters, e.g., in the digital library community. *Functionality* is supported by an XML expression *outside* the package.

What should be in this structural metadata, whether inside or out? Most of the user communities named above seek not only a list of the content elements--like the directory list in a zip or tar file--but also information about how the listed elements relate to one another and about their provenance. There is also interest in data-integrity support, e.g., hash values aka checksums or the equivalent.

Here are a few instances when this structural metadata impulse has emerged as a specification:

- **METS** from the digital library community,[5] generally used to create virtual packages (metadata in one file, content objects in other files). Supports user-interface presentations and other complex functions.
- **Bag-It** from the digital library community,[6] also generally used to create virtual packages, for delivery or storage, but much simpler to create (and less richly functional) than METS.
- **IMF Packing List**, from version 1.0 of the Interoperable Master Format specification.[7] The IMF Packing List is defined as, "A list describing the files and providing a means for authentication of the files as delivered in a package."
- **MPEG-21 DID** Model (Digital Item Declaration Model) as described in the MPEG-21 set of standards, ISO/IEC 21000-2, *Information technology — Multimedia framework (MPEG-21) — Part 2: Digital Item Declaration*, 2003. Digital Items are defined as structured digital objects, including a standard representation and identification, and metadata.

Closer at hand, in our MXF-and-video neighborhood, there is the example of the *manifest file* in the AS-02 specification (*MXF Versioning*), still in draft form. The metadata "in" an MXF file is encoded as KLV (Key-Length-Value) while the "associated but separate" AS-02 manifest is encoded in XML. The draft AS-02 specification defines the manifest as "a list of all of the files and folders in the bundle, including the version and essence component files, with their identifiers and relationships. The manifest file shall also contain a unique identifier that shall provide the identification of the bundle as a whole."

We propose using this useful concept and specification in AS-AP. The trend of our discussion has been to consider it to be an optional element. In any case, it ought not include any information that is not also in the MXF file for long-term management. Oliver Morgan said that

[5] http://www.loc.gov/standards/mets/.
[6] http://tools.ietf.org/html/draft-kunze-bagit-06.
[7] http://www.etcenter.org/wp-content/uploads/mailchimp-downloads/IMF_Specification_V1.0.pdf, February 2011.

there will always be additional information inside the MXF file that is not particularly relevant to manifest operations. Our working principle, he added, should be "you can always derive the manifest from the MXF header and you can always start from a manifest and upgrade it into an MXF header. We must avoid allowing the two schemas to form an overlapping Venn diagram."

Assuming that AS-02 will be published soon, our draft wording in AS-AP is:

> Manifest. AS-AP files may be accompanied by a manifest that provides an inventory of the file's essences and expresses the relationships between them, as well as providing summary information about the essence item and its provenance. This manifest shall follow the specification in section 9 of AS-02 (*MXF Versioning*).

What's in AS-02? In AS-02, content assets take the form of a bundle of more or less atomic MXF files, in single-essence OP1a form. In your media "factory" you can assemble the assets as needed to produce one or another version of the underlying work. The introduction to the draft AS-02 specification says, "The goal of the AS-02 work was to create a factory format that supports multiple versions and with no reinvention, making use of metadata-only MXF (SMPTE 377-1-2009) to provide component synchronization."

The *manifest file* is spelled out in section 9:

> The AS-02 manifest file shall be encoded as XML. This file shall include the following mandatory elements: BundleName (contains the file name of the root-level folder of the bundle), BundleID (identifier as URN-encoded UUID), Creator (free-form, human-readable annotation), CreationDate (time and date at which the bundle was created), and FileList (list of File elements that describe the files and folders in the bundle). The manifest may also contain the optional AnnotationText (free-form, human-readable annotations).
>
> For each file in the FileList, there shall be a FileType element that includes the following mandatory elements: FileID (identifier as URN, three possible sources), Role (how the file is used), Size (size of file in bytes, encoded as type xs:positiveInteger), and Path (URI of the file, relative to the root of the bundle). Each FileType may also contain the optional MIC (media integrity check value for the file; of the type xs:hexBinary; may take the form of HMAC-SHA1, crc32, crc16, or md5; with scope indicated: essence_only or entire_file), and AnnotationText (free-form, human-readable annotations).

You may ask, "Why don't you use AS-02 instead of writing your new AS-AP application specification?" As noted earlier, for preservation, we are drawn to the full-content encapsulation offered by a single file (although one could of course concoct a method to encapsulate the AS-02 bundle). In addition, with an eye on our regular work of videotape reformatting, we foresee that the best essence arrangement will be as frame-wrapped and interleaved. Interleaved (i.e., not atomic) essences are forbidden in AS-02.

Metadata[8]

Our Working Group discussions have highlighted a dual approach to the embedding of metadata in MXF AS-AP files. On the one hand, we have talked about a *minimal header element*, including a minimalist DM (Descriptive Metadata) track tailored for AS-AP. In spirit, this header-cum-DM-track would be akin to the Broadcast Extension (bext) chunk in the Broadcast WAVE file standardized by the European Broadcast Union. In practice, we would like to see something less constrained, with room for a bit more data, not the least of which would be multiple tagged identifiers. The data in the header-cum-DM-track would be basic, consisting of metadata needed to identify and manage the content object as an object, e.g., in a preservation asset management system. In addition to one or more identifiers, another important metadata element is a statement of responsibility, sometimes called attribution, to identify the organization that serves as the preservation "keeper" for this content item.

On the other hand, to support *more complete representations* of descriptive, administrative, and technical metadata, we are discussing finding places "deeper in the file." Oliver Morgan has called our attention to MXF Generic Stream Partitions, standardized in SMPTE ST 410-2008. These partitions were established for a variety of applications, one of which was to contain various classes of data streams (including text streams), such as extensive blocks of "metadata that cannot suitably be stored in the Header Metadata (e.g., specialized preservation metadata)." A corollary Recommended Practice document from SMPTE is RP 2057-2011, titled Text-Based Metadata Carriage in MXF, which "defines how to carry text-based metadata with a specified text MIME type encoded using either Unicode UTF-8 or UTF-16 character encoding (such as XML) in a MXF file."

What types of metadata might be placed in Generic Partitions, and about which types might the Working Group offer recommendations? We are certainly interested in technical metadata, the moving image equivalents to a pair of standards from the Audio Engineering Society: (1) AES57 (forthcoming; the draft form was labeled AES X098B), titled *AES standard for audio metadata - Audio object structures for preservation and restoration* and (2) AES-X098C (still only in draft), titled *Administrative metadata for audio objects - Process history schema*. The former provides a description of a given file's technical characteristics, not unlike the instantiation elements in PBCore, while the latter offers a description of the process that created the file, related to what is sometimes called digital provenance.

In contrast, although we acknowledge the importance of descriptive metadata, we may not offer much in the way of specific recommendations. The Working Group includes representatives from both archive and library organizations, with practices for resource description that vary in significant ways. With all types of materials, libraries favor bibliographic data while archives prefer finding aids. In the end, we are likely to have more to say about where a chunk of agency-produced descriptive metadata might be embedded than about what it should look like.

We also do not anticipate offering recommendations regarding structural metadata, beyond the structural options inherent in MXF itself. Different agencies and even units within agencies take

[8] This section of this document is based on one portion of an article I wrote for the July 2011 issue of the *IASA Journal* (number 37, pp. 39-40). Copies of this and other issues of this valuable journal may be obtained via http://www.iasa-web.org/iasa-journal.

a variety of approaches to content packaging, i.e., the binding or bundling of multiple related files. We have, however, discussed the idea of having MXF AS-AP include a way to wrap collections, i.e., sets of items.