Elias Oltmanns[1], Tim Hasler[2]

Outline of an archiving strategy for digital audiovisual material

[1] 0000-0003-1453-7063
[2] 0000-0001-9164-3500

ZIB Report 20-22 (July 2020)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: `bibliothek@zib.de`
URL: `http://www.zib.de`

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Outline of an archiving strategy for digital audiovisual material

Elias Oltmanns, Tim Hasler (Zuse Institute Berlin)

July 8, 2020

Contents

This is some preliminary documentation on current results of a research project jointly conducted by Stiftung Deutsche Kinemathek (SDK)[1] and Zuse Institute Berlin (ZIB)[2]. In this project, we are working on a practical yet sustainable archiving solution for audiovisual material. As it turned out, we had to tackle two major obstacles:

1. Metadata is collected according to standards established in the community but lacking a prescribed serialisation format.
2. Storage size of audiovisual material and time scales of production processes make it often impractical to defer submission for archival storage until all components have arrived and can be processed in one go.

In order to address the first issue, we have been working on a mapping from EN 15744[3] and some metadata from the FIAF Cataloguing Manual[4]

[1] https://www.deutsche-kinemathek.de/en/home
[2] https://www.zib.de/
[3] http://filmstandards.org/fsc/index.php?title=EN_15744
[4] https://www.fiafnet.org/pages/E-Resources/Cataloguing-Manual.html

to EBUCore[5], which can be serialised in XML and therefore embedded into METS[6]. For the second part, we aim at establishing a Persistent Identifier (PID) system with suitable metadata schema and public resolver jointly run by the film archiving community.

Even though it is still work in progress, we would like to share current results with the community. Interested parties can find information and supplementary material on the project webpages at the nestor (the german network of expertise in long-term preservation) working group on audiovisual content (also check the documents section)[7]. The present article is meant to outline our ideas in three sections: Firstly, we provide a slightly more extensive motivation for our overall strategy. Secondly, we explain the reasoning behind our choice of metadata formats. Finally, we sketch the current state of and future ideas for our METS/EBUCore implementation guidelines for archiving audiovisual material. Feedback and contributions from the community are welcome.

1 Motivation and general approach

Producing a digital restoration of audiovisual master material, for instance, involves a lot of resources in terms of both, data storage and processing time. In order to document the process, it is desirable to archive not only the final distributable files but also intermediate stages leading up to them and (digital / digitised) source material they have been derived from. Conversely, adequate knowledge about the whole context may be helpful or even imperative in order to make sense of individual parts. Archiving the whole lot in a single package may be undesirable or even unfeasible due to resource constraints. According to the Reference Model for an Open Archival Information System (OAIS)[8], however, an Archival Information Package (AIP) should be independently understandable by “the designated community”. Therefore, we try to develop an archiving strategy that acknowledges the need for splitting material into manageable pieces while providing adequate mitigation when this yields AIPs that might not be fully comprehensible without reference to other parts of the material. Also, the occasional minor update to certain metadata would ideally not involve (re)processing huge AIPs with otherwise unchanged contents.

In our estimation, a Persistent Identifier (PID) system with suitable metadata associated with each PID can be a very powerful tool in such an archiving strategy. In our scheme, it is supposed to serve two key purposes:

- Given two AIPs, members of the designated community can easily find out (without consulting external sources) whether they have been generated as part of the same archival version of some cinematographic work or not.

[5] https://tech.ebu.ch/docs/tech/tech3293.pdf
[6] https://www.loc.gov/standards/mets/
[7] https://wiki.dnb.de/display/NESTOR/Persistent+Identifiers+for+Audio+Visual+Heritage
[8] http://www.oais.info/

- Anyone presented with an AIP with insufficient context for a full understanding of its contents can turn to a public resolver with the supplied PIDs in order to find out about other AIPs generated as part of the same archival version of a cinematographic work.

For the purposes of our project, the (archival) "version" of audiovisual material is intended to provide as much information as required (and available at the time) for the designated community to make sense of it independently of other sources. A version can be split into manageable pieces called "dataObjects" for archival storage in order to handle the size or delays in production of different components. The archiving strategy involves assigning a PID to the version as a whole and to each dataObject that constitutes a part of it. The metadata associated with the PID for the version should always contain a complete list of the PIDs of related dataObjects that have been submitted for archival storage. Conversely, the metadata associated with the PID of any dataObject should always contain the PID of the related version.

In practical terms, we suggest to assign a PID with a core set of metadata to a version first and subsequently to related dataObjects as they become available for submission. In the interest of linked data, we suggest that the metadata of versions should include a PID for the cinematographic work subject to the version of audiovisual material.

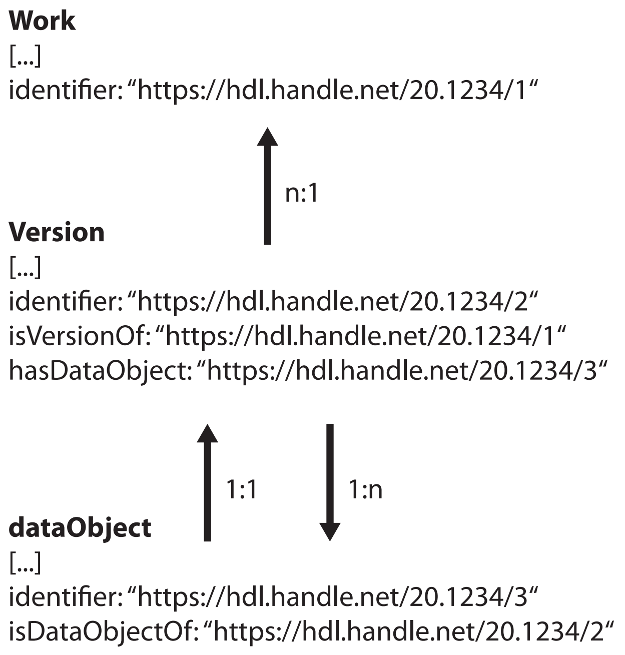
Work
[...]
identifier: "https://hdl.handle.net/20.1234/1"
n:1
Version
[...]
identifier: "https://hdl.handle.net/20.1234/2"
isVersionOf: "https://hdl.handle.net/20.1234/1"
hasDataObject: "https://hdl.handle.net/20.1234/3"
1:1
1:n
dataObject
[...]
identifier: "https://hdl.handle.net/20.1234/3"
isDataObjectOf: "https://hdl.handle.net/20.1234/2"

Figure 1: Relations of PIDs

Whereas PIDs of versions and their dataObjects are solely the content provider's (source organisation's) responsibility, procedures should be in place to encourage shared use of PIDs for cinematographic works and ensure trustworthy maintenance of the associated metadata. This might conceivably involve external sources similar to the GND for Cultural Data (GND4C)[9] or an authority or review based registration process within the PID system discussed here. Also, when a version has been cooperatively produced by different parties who want to archive their own copy of that version, all parties should register and maintain individual PIDs for version and dataObjects but use the “sameAs” field in the metadata of their version PID in order to list the corresponding version PIDs registered by their partners.

Thus, key functional requirements for the PID system include:

- version metadata should include a complete list of PIDs for all dataObjects associated with the version.
- dataObject metadata should include the PID of the version (or possibly multiple versions) it is associated with.
- Registering a new version in the PID system should not require updating the metadata record of the associated work.
- All metadata sets should indicate the source and when they have been last updated.

This implies that version metadata must be updated each time a PID is assigned to a related dataObject, whereas metadata of the work as well as other related dataObjects is unaffected by such an operation.

Using the established metadata standards METS and EBUCore (see next section for details), an OAIS could facilitate the PID system in the following way:

- Prior to ingest, dataObjects are required to have a PID assigned to them which is associated with a version, which in turn is associated with a work. Also, for each of these three PIDs, a core set of metadata, suitable for the respective category, is registered within the system.
- Exactly one archival information package (AIP) is created per dataObject.
- The METS in each AIP contains separate EBUCore documents recording the metadata registered in the PID system for work, version, and the dataObject included in the AIP.
- In particular, the EBUCore document representing the version will typically contain references to other dataObjects.

While current metadata in the PID system is authoritative, we would still suggest to provide a full record of all relevant metadata in an AIP. Naturally, the METS in an AIP can only record a snapshot of the metadata registered with the respective PIDs at a certain point in time. Therefore, a lastModified timestamp should be part of

[9] https://www.dnb.de/EN/Professionell/ProjekteKooperationen/Projekte/GND4C/gnd4c.html

the metadata schema and updated accordingly. In the AIP, this is documented in the @lastModified{Date,Time} attributes of the <ebuCoreMain> element.

In principle, METS files from the AIPs of any two dataObjects associated with the same version are intended to coincide on the embedded EBUCore documents representing the work and version metadata from the PID system. Conflicts will arise, however, especially when archiving the first dataObjects of a given version before all remaining dataObjects belonging to the same version have been assigned a PID. On dissemination, such conflicts can optionally be resolved by updating all metadata records to the current state in the PID system (if accessible) or by duplicating metadata (only for work and version) from the AIP with the newest lastModified timestamp on the respective <ebuCoreMain> element. An archive may be well advised to record metadata for work and version in an abstract archival information collection (AIC) and update it as appropriate on ingest of new dataObjects. This will aid the implementation of tools for discovery as well as the conflict resolution algorithm outlined above.

1.1 A side note on discovery

Since we will have a lot of linked metadata available in the PID System from hopefully many participating institutions, it could also serve as a basis for some kind of union catalogue. The basic assumption is that if a user wants to know something about e.g. Fritz Langs „Metropolis“, the first step would be to just type the title in some google-like search field. The first result should be a synopsis of the work, maybe associated with a picture (a still image or a film poster). Because the work is an abstract entity, its descriptive metadata should be the same for all the participating archives. One source for this consolidated synopses could be the aforementioned GND4C.

As you have seen in figure 1 (page 3) the relations of the various parts of a cinematographic work are expressed via references in the PID system. Based on that information the presentation will be handled by the user interface. You may have noticed that there is a linkage only from the actual version to the work it manifests and not vice versa. So the logic behind the user interface for the search has to look up which versions link to a specific work and display them on the results page for that work. This has on the one hand the advantage that the metadata for the work stays untouched once it is fed into the PID system and on the other hand that we are able to accumulate many versions of that work as we go along.

On that results page a list of all available variants registered in the PID system will be shown. So even if there are two variants represented in identical digital objects, they may differ in the physical place where they are stored. The user could decide which institution to approach by location or personal preference.

We have to identify various potential use cases: There could be someone interested in just finding a streamable version of a film, whereas others may be interested in a particular resource that is available to that film. We suggest to have all three levels down to the technical metadata describing the digital object available for presentation. The emphasis is here on available as in contrast to unavoidable. Since the possible usergroup comprises not necessarily only archivists, it is good to have the data for a faceted search available but not to overwhelm the interested layman on the other hand. But if a user is interested in the version of Metropolis kept at the SDK, a link to the SDK can be provided parallel to a possibly quite extensive list of associated digital objects. In our scenario even someone looking for the transcribed subtitles of a specific film can see if and where they are available and negotiate with the holding institution over the possibility to use them.

In a follow up paper on the progress of the project we will look into the discovery tool more closely.

2 Considerations regarding metadata

On submission of a dataObject for archival storage, PIDs and associated metadata of the dataObject itself, its version and the associated work should be included as descriptive metadata. We suggest to use the Metadata Encoding and Transmission Standard (METS)[10] as the road map to an AIP. It is an established standard suitable for archiving diverse data and has the means to provide (among other things):

- Structural information relevant for the interpretation of the package contents and / or its context.
- A complete record of the files included in the package accompanied with integrity and media type information.
- A provenance section for the archive to record preservation actions (typically in PREMIS[11]).
- Container elements to include domain specific metadata (both technical and descriptive).

Moreover, metadata can be included from external files in various ways. This also provides the possibility to split a potentially very large METS file into a set of smaller files hierarchically linked to a parent METS file.

As for domain specific metadata, we faced the problem that no generally accepted serialisation formats exist for established standards and models in the film archiving community like metadata for Film identification (EN 15907)[12] or the FIAF Moving Image Cataloguing Manual[13]. Instead of implementing a complete serialisation of these rather complex models, we decided to restrict ourselves to the Minimum set of metadata for cinematographic works (EN 15744)[14] augmented by a selection of metadata from the aforementioned FIAF Moving Image Cataloguing Manual. Moreover, we decided to define a mapping onto an existing data model specifically designed for the exchange of metadata

[10]https://www.loc.gov/standards/mets/
[11]https://www.loc.gov/standards/premis/
[12]http://filmstandards.org/fsc/index.php?title=EN_15907
[13]https://www.fiafnet.org/pages/E-Resources/Cataloguing-Manual.html
[14]http://filmstandards.org/fsc/index.php?title=EN_15744

describing audiovisual contents: EBUCore[15]. Since METS as the container format has only been developed as an XML schema, we focus on the XML schema of EBUCore[16] here. Nevertheless, the EBUCore ontology[17] might prove useful in the context of the PID system and linked data.

Work in progress results of our mapping effort are provided as spreadsheets in the documents section at our project webpage[18]. EBUCore elements are specified in a somewhat shortened XPath notation. Note, however, that there are several EBUCore documents embedded into the METS file. Column headings in the spreadsheet indicate where subsequent elements are supposed to be found in the METS file.

Only a subset of the domain specific metadata is intended to be recorded within the PID system. Details are yet to be determined as the mapping matures but there is one consideration of a more general nature: Metadata directly accessible through the PID system could be made available for linked data and as a discovery aid. This is a compelling reason for recording metadata within the PID system that helps members of the community to identify versions and dataObjects they might be interested in. A lot, if not all, of the metadata associated with the cinematographic work and the version will fall into this category. On the other hand, metadata that is mostly required for a correct interpretation but not the discovery and identification of certain audiovisual material is an essential part of the AIP but less so within the PID system. It is our hypothesis that this applies to the vast majority of technical metadata associated with a dataObject and that modifications to this metadata happen to be particularly unlikely without modifications to contents of the dataObject itself.

Assuming that subsequent modifications of metadata are most likely to be associated with a cinematographic work or a version rather than a particular dataObject, an archive can record this in terms of an Archival Information Collection (AIC) mentioned in the previous section. This makes it a lightweight operation since updating AIPs containing the associated dataObjects is not strictly necessary due to the merging algorithm described earlier.

3 METS file: a road map to the AIP

A sample METS file demonstrating the guidelines and recommendations discussed in this section is available from the documents section at our project webpage[19]. Also, see the METS Primer[20] and current METS schema documentation[21] provided by the Library of Congress for detailed definitions of elements mentioned below.

[15]https://tech.ebu.ch/docs/tech/tech3293.pdf
[16]https://www.ebu.ch/metadata/schemas/EBUCore/ebucore.xsd
[17]https://www.ebu.ch/metadata/ontologies/ebucore/
[18]https://wiki.dnb.de/display/NESTOR/Persistent+Identifiers+for+Audio+Visual+Heritage
[19]https://wiki.dnb.de/display/NESTOR/Persistent+Identifiers+for+Audio+Visual+Heritage
[20]https://www.loc.gov/standards/mets/METSPrimer.pdf
[21]https://www.loc.gov/standards/mets/mets-schemadocs.html

3.1 Context: semantic structure with associated metadata

`<structMap>` is the only mandatory top-level element in a METS file. It is repeatable and used to record all sorts of hierarchical structures deemed relevant for a proper interpretation of the described contents. These structures are represented by nested `<div>` elements. Metadata as well as files can optionally be associated with them via appropriate references to other sections of the METS file.

We suggest two `<structMap>`s distinguished by their `@TYPE` attribute. One is required in order to relate the dataObject to its version and cinematographic work and could even be extended to capture certain internal structures (e.g. image sequences). Another one is suggested in order to record the filesystem structure of the whole package at the time of submission to the agent responsible for its long-term preservation (see next subsection).

At the time of this writing, we have implemented the first `<structMap>` in a very straightforward way. It simply reflects the hierarchical relationship between cinematographic work, version and dataObject and lists all files of the package as direct children of the dataObject. It would roughly look like this:

```
<mets xmlns="http://www.loc.gov/METS/">
  <structMap TYPE="logical">
    <div DMDID="metadataFromWorkPID" ID="LOG_0000"
         LABEL="${titleOfWork}" TYPE="cinematographicWork">
      <div DMDID="metadataFromVersionPID" ID="LOG_0001"
           LABEL="${titleOfVersion}" TYPE="version">
        <div ADMID="techMDOfDataObject sourceMDOfDataObject"
             DMDID="DMDLOG_0002" ID="LOG_0002"
             LABEL="${titleOfDataObject}" TYPE="dataObject">
          <div>
            <fptr FILEID="FILE_0001_MASTER" />
          </div>
          <div>
            <fptr FILEID="FILE_0002_MASTER" />
          </div>
          <!-- ... -->
        </div>
      </div>
    </div>
  </structMap>
</mets>
```

In this `<structMap>`, the first three `<div>` elements are expected to occur exactly once in that order with values of the `@TYPE` attribute as specified. Additionally, EBUCore representations of the metadata associated with the PIDs for work, version and dataObject are referenced via the `@DMDID` attribute. Make sure that these EBUCore documents are marked as originating from the PID system, so they can be distinguished from other descriptive metadata that might be added to the `@DMDID` attribute. The

`@documentLocation` or `@typeLabel` attribute of the `<ebuCoreMain>` element might be suitable candidates for that purpose.

Please note that references to the included files are provided as a flat list right now. Filesystem hierarchy is intentionally not reflected here, we will come to that in the next subsection. METS does provide additional elements that may help to extend the logical `<structMap>` in very interesting ways, though. We have not had resources to look into that properly yet, but here is a taste of what it might look like:

```
<mets xmlns="http://www.loc.gov/METS/">
  <structMap TYPE="logical">
    <div DMDID="metadataFromWorkPID" ID="LOG_000000"
         LABEL="${titleOfWork}" TYPE="cinematographicWork">
      <div DMDID="metadataFromVersionPID" ID="LOG_000001"
           LABEL="${titleOfVersion}" TYPE="version">
        <div ADMID="techMDOfDataObject sourceMDOfDataObject"
             DMDID="DMDLOG_0002" ID="LOG_000002"
             LABEL="${titleOfDataObject}" TYPE="dataObject">
          <div TYPE="dcdm">
            <fptr>
              <par>
                <seq LABEL="image sequence">
                  <area FILEID="FILE_000000_MASTER" LABEL="image 1"
                        ORDER="1" />
                  <area FILEID="FILE_000002_MASTER" LABEL="image 2"
                        ORDER="2" />
                  <!-- ... -->
                </seq>
                <area FILEID="FILE_090000_MASTER" LABEL="audio" />
                <area FILEID="FILE_090001_MASTER" LABEL="subtitles" />
                <area FILEID="FILE_090002_MASTER" LABEL="playlist" />
                <!-- ... -->
              </par>
            </fptr>
          </div>
        </div>
      </div>
    </div>
  </structMap>
</mets>
```

3.2 What about files then?

Files are generally not referenced directly from the `<structMap>`. Instead, a METS file typically includes the `<fileSec>` element listing all files in the package and assigning a unique identifier to each of them which can be referenced from the `<structMap>`. Although `<fileGrp>` elements offer the possibility to organise contents of the `<fileSec>` into all sorts of (nested) sets, it is important to realise that the `<fileSec>` is, per se, volatile. Files may be added at any time due to preservation actions and, in fact, there is no guarantee that the directory structure of existing files is preserved over time. Therefore it is good practice to record the structure of the filesystem hierarchy in a separate `<structMap>` element, so the original files, at least, can always be restored to a directory structure that is known to have been working before ingest into the archive. Relative links between those files are an obvious reason for preserving directory hierarchies. We therefore suggest an additional `<structMap>` along the following lines:

```
<mets xmlns="http://www.loc.gov/METS/">
  <structMap TYPE="filesystemAtSubmission">
    <div ID="FS_0000" LABEL="1313315-D" TYPE="directory">
      <div ID="FS_001" LABEL="190719_OV" TYPE="directory">
        <div ID="FS_0002"
             LABEL="FuckingCity_FTR-1-25_F-137_DE-XX_DE-NR_51_2K_NULL..."
             TYPE="directory">
          <div ID="FS_0003"
               LABEL="140e5068-c86f-471b-8965-5c2598b10f85_j2c.mxf"
               TYPE="item">
            <fptr FILEID="FILE_0001_MASTER" />
          </div>
          <div ID="FS_0004"
               LABEL="1da6204a-d232-4747-b168-b7027caf1f30_j2c.mxf"
               TYPE="item">
            <fptr FILEID="FILE_0002_MASTER" />
          </div>
          <!-- ... -->
        </div>
      </div>
      <div ID="FS_0014" LABEL="190722_VF_UTen_UTfr" TYPE="directory">
        <div ID="FS_0015"
             LABEL="FuckingCity_FTR-1-25_F-137_DE-EN_DE-NR_51_2K_NULL..."
             TYPE="directory">
          <div ID="FS_0016" LABEL="ASSETMAP" TYPE="item">
            <fptr FILEID="FILE_0012_MASTER" />
          </div>
          <!-- ... -->
        </div>
      </div>
    </div>
  </structMap>
</mets>
```

As a rule of thumb, we suggest to leave this `<structMap>` as pure as possible and assign descriptive metadata at an appropriate level in the logical `<structMap>` described in the previous section. Administrative metadata in METS terminology may be assigned either in the logical `<structMap>` too, or directly in the `<fileSec>`.

Concerning the `<fileSec>`, two requirements are particularly essential in the archiving context: All values of `@xlink:href` attributes need to be relative paths and integrity information needs to be supplied for all `<file>` elements in accordance with the METS schema. Note that the same is true for the `<mptr>` element in a `<structMap>` or a `<mdRef>` element in one of the metadata sections if you should venture to employ one of those. We also recommend media type information as demonstrated in the sample METS file at our project webpage.

3.3 Designated places for metadata

See section "Considerations regarding metadata" above for information about our effort to map a subset of metadata from established models in the film archiving community onto EBUCore. This mapping has been specified in terms of formally independent EBUCore documents that are to be found in different locations across the METS file by consulting the logical `<structMap>`. It needs to be highlighted, however, that decisions as to what element should go to which EBUCore document are partly but not exclusively based on the semantics of METS elements `<dmdSec>`, `<techMD>`, and `<sourceMD>`. Once getting a PID system as described in the first section operational, we consider it a guiding principle to align metadata schemas in such a way that metadata retrieved from the PID system can be matched exactly onto one `<dmdSec>`, respectively, for work, version and dataObject at the time of creating the METS file.

Another, potentially conflicting, principle is to avoid data duplication between metadata in a `<techMD>` section associated with the dataObject and the `<dmdSec>` associated with the dataObject's PID. This should only occur if the metadata field is considered essential in the PID system, yet also provides technical information estimated to be of practical use supporting (automated) quality assurance in an event of format migration or similar.

There is generally no limit to the number of `<dmdSec>`s, `<techMD>`s, `<sourceMD>`s, or, indeed, `<rightsMD>`s, and `<digiprovMD>`s associated with elements in the logical `<structMap>` or, under certain circumstances, the `<fileSec>`. That is, why it is important to make the `<dmdSec>`s originating from the PID system stand out in order to resynchronise them as appropriate. See subsection "Context: semantic structure with associated metadata" for viable options.

4 Acknowledgements

Results presented in this document have been cooperatively developed by a project team consisting of Antje Dittmann, Annette Groschke, Niklas Hütter, Klaus Suttner, and Maxi Zimmermann at Stiftung Deutsche Kinemathek, Johannes Starlinger as an external developer and the authors at ZIB. On behalf of the team, we also would like to thank Jean-Pierre Evain at European Broadcasting Union for his many contributions and valuable advice as developer and mainainer of EBUCore.