METS as an Intermediary Schema for a Digital Library of Complex Scientific Multimedia

Richard Gartner

ABSTRACT

The use of the Metadata Encoding and Transmission Standard (METS) schema as a mechanism for delivering a digital library of complex scientific multimedia is examined as an alternative to the Fedora Content Model (FCM). Using METS as an "intermediary" schema, where it functions as a template that is populated with content metadata on the fly using Extensible Stylesheet Language Transformations (XSLT), it is possible to replicate the flexibility of structure and granularity of FCM while avoiding its complexity and often substantial demands on developers.

METS as an Intermediary Schema for a Digital Library of Complex Scientific Multimedia

Of the many possible approaches to structuring complex data for delivery via the web, two divergent philosophies appear to predominate. One, exemplified by such standards as the Metadata Encoding and Transmission Standard (METS)[1] or the Digital Item Declaration Language (DIDL),[2] relies on the structured packaging of the multiple components of a complex object within "top-down" hierarchies. The second, of which the Fedora Content Model (FCM) is perhaps a prime example,[3] takes the opposite approach of disaggregating structural units into atomistic objects, which can then be recombined according to the requirements of a given application.[4] Neither is absolute in its approach—METS, for instance, allows cross-hierarchy linkages, and many FCM models are designed hierarchically—but the distinction is clear.

Many advantages are validly claimed for the FCM approach to structuring digital data objects. Individual components, not constrained to hierarchies, may be readily reused in multiple representations with great flexibility.[5] Complex interobject relationships may be encoded using semantic linkages,[6] a potentially much richer approach to expressing these than the structural relationships of XML can allow. Multiple levels of granularity, from that of the collection as a whole down to its lowest-level components, can readily be modelled, allowing interobject relationships to be encoded as easily as intercomponent ones.[7]

Such models, particularly the RDF-based Fedora content model, are very powerful and flexible, but can often lead to complexity and consequently considerable demands on system development before they can be implemented. In addition, despite the theoretical interoperability offered by RDF, in practice the exchange and reuse of content models has proved somewhat limited because considerable work is usually required to re-create and validate a content model created elsewhere.[8]

This article examines whether it is possible to replicate the advantages of this approach to structuring data within the constraints of the more rigid METS standard. The data used for this analysis is a set of digital objects that result from biological nanoimaging experiments, the interrelationships of which present complex problems when they are delivered online. The

Richard Gartner (richard.gartner@kcl.ac.uk) is a Lecturer in Library and Information Science, King's College, London.

method used is an unconventional use of a METS template as an intermediary schema;[9] this allows something of the flexibility of the FCM approach while retaining the relative simplicity of the more structured METS model.

A Nanoimaging Digital Library and its Metadata Requirements

The collection analysed for this study derives from biological nanoimaging experiments undertaken at the Randall Division of Cell and Molecular Biophysics at King's College London. Biological nanoimaging is a relatively new field of research that aims to unravel the biological processes at work when molecules interact in living cells; this is done by using optical techniques that can resolve images down to the molecular level. It has particular value in the study of how diseases progress and has great potential to help predict the effects of drugs on the physiology of human organs.

As part of the Biophysical Repositories in the Lab (BRIL) project at King's College London,[10] a digital library is being produced to meet the needs of practitioners of live cell protein studies. Although the material being made available here is highly specialised, and the user base is restricted to a specialist cohort of biologists, the challenges of this library are similar to those of any collection of digital objects: in particular, the metadata strategy employed must be able to handle the delivery of complex, multifile objects as efficiently as, for example, a library of digitized books has to manage the multiplicity of image files that make up a single digital volume.

The digital library itself is hosted on the widely used Fedora repository platform; as a result, it is employing FCM as the basis of its data modelling. The purpose of this analysis is to ascertain whether METS can also be used for the complex models required by this data and to compare its potential viability as an architecture for this type of application with FCM.

A particular challenge of this collection is that the raw images from which it is constituted require combining and processing before they are delivered to the user. A further challenge is that the library encompasses images from a variety of experiments, all of which combine these files in different ways and employ different software for processing them. Some measure of the complexity of these requirements can be gathered from figure 1 below, which illustrates the processes involved in delivering the digital objects for two types of experiments.

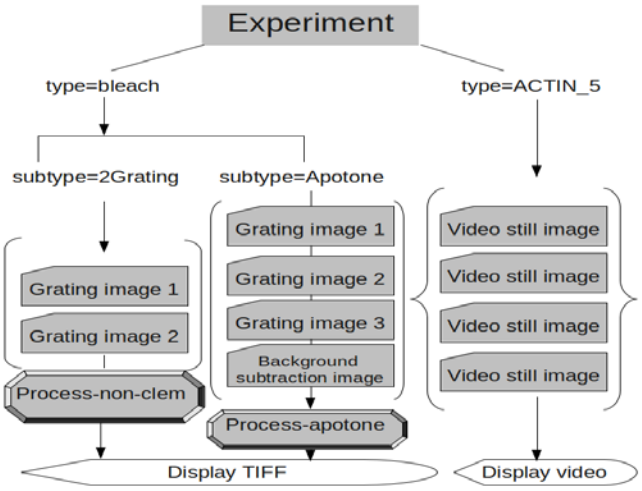

Figure 1. Architecture for Two Experiment Types

The images created by two experiments, *bleach* and *ACTIN_5*, are shown here: it will be seen that the *bleach* experiment is divided into two subtypes (here called *2Grating* and *Apotone*). Each type or subtype of experiment has its own requirements for combining the images it produces before they are displayed.

For the subtype *2Grating*, for instance, two images, each generated using a different camera grating, are processed in parallel (indicated by the brackets); these are then combined using the software package *process-non-clem* (shown by the hexagonal symbol) to produce a display image in TIFF format. The subtype *Apotone* requires three grating images and a further image with background information to be processed in parallel by the software *process-apotone*; in this case, the background image provides data to be subtracted from the combined three grating images to produce the final TIFF for display. *ACTIN_5* experiments are entirely different: they produce still images that need to be processed sequentially (shown by the braces) to produce a video.

Encoding the BRIL Architecture in METS

This architecture, although complex, is readily handled within METS in a manner analogous to that of more conventional collections. As in any METS file, the structure of the experiments, including their subexperiments, is encoded using nested division (* *) elements in the structural map (example 1a).

```
<div TYPE= “experiment-types”>
<div TYPE= “experiment-bleach” ID= “experiment-type0001”>
<div TYPE= “experiment-bleach-2Grating” ID= “experiment-type0001-subtype001”>
 [subsidiary <div>s containing image information]
</div>
<div TYPE= “experiment-bleach-apotone” ID= “experiment-type0001-subtype002”>
 [subsidiary <div>s containing image information]
</div>
</div>
<div TYPE= “experiment-actin_5” ID= “experiment-type0002”>
 [subsidiary <div>s containing image information]
</div>
</div>
```

Example 1a. Sample Experiment-Level Structural Map

Within these containing divisions, subsidiary * * elements are used to map the combination of images necessary to deliver the content for each type. METS allows the specification of the parallel or sequential structuring of files using its *<par>* and *<seq>* elements respectively. The parallel processing of the *Apotone* subtype, for instance, could be encoded as shown in example 1b.

```
<div TYPE= “experiment-bleach-apotone-images” ID= “apotone-000”>
<fptr>
 <par>
 <area FILEID= “apotone-grating1–000”/>
 <area FILEID= “apotone-grating2–000”/>
 <area FILEID= “apotone-grating3–000”/>
 </par>
</fptr>
</div>
```

Example 1b. Sample Parallel Structure for Raw Image Files to be Combined Using a Process Specified in Associated Metadata (Behavior Section)

Each division of the structural map of this type may in its turn be attached to a specific software item in the METS behavior section to designate the application through which it should be processed: the tri-partite set of images in example 1b, for instance, would be linked to the *process-apotone* software using the code in example 1c.

```
<behavior GROUPID= “process-apotone” STRUCTID= “apotone-000”>
 <mechanism LOCTYPE= “URL” ns2:href= “software/process-apotone”/>
</behavior>
```

Example 1c. Sample METS Behavior Mechanism for a Specification of Image Processing

This approach is straightforward, and METS is capable of encoding all of the requirements of this data model, although at the cost of large file sizes and a degree of inflexibility. This may be no problem when the principle rationale behind the creation of this metadata is preservation: linking all of the project metadata in a coherent, albeit monolithic, structure of this kind benefits especially its usage as an Open Archival Information System (OAIS) Archival Information Package (AIP), one of the key functions for which METS was designed. Problems are likely to arise, however, when this approach is scaled up in a delivery system to include the potentially millions of data objects that this project may produce.

The large size of the METS files that this approach necessitates makes their on-the-fly processing for delivery much slower than a system that uses aggregations of the smaller files required by the FCM model and so processes only metadata at the granularity necessary for the delivery of each object. Such flexibility is much harder to achieve within METS, although mechanisms that currently exist for aggregating diverse objects within METS may seem to offer some degree of solution to this problem.

Complex Relationships under METS

Underlying the METS structural map is an assumed ontology of digital objects that encodes a long-established view of text as an ordered hierarchy of content objects;[11] this model accounts for the

map's use of hierarchical nesting and the ordinality of the object's components. The rigidity of this model is alleviated to some extent by the facility within METS to encode structural links that cut across these hierarchies. These links, which join nodes at any level of the structural map, are particularly useful for encoding hyperlinks within webpages,[12] and so are often used for archiving websites.

Various attempts have been made to extend the functionality of the structural map and structural links sections to allow more sophisticated aggregations and combinations of components beyond the boundaries of a single digital object, in a manner analogous to the flexible granularity of FCM. METS itself offers the possibility of aggregating other METS files through its *<mptr>* (METS pointer) element: this element, always a direct child of a * * element in the structural map, references a METS file that contains metadata on the digital content represented by this * *. For example, two complex digital objects could be represented at a higher collection level, as shown in example 2.

```
<div>
<div ID= "bleach2grating-1">
<mptr LOCTYPE= "URL" xlink:href= "twogratingbleach6-grating1–01.xml"/>
</div>
<div ID= "bleach2grating-2">
<mptr LOCTYPE= "URL" xlink:href= "twogratingbleach6-grating1–02.xml"/>
</div>
</div>
```

Example 2. Use of METS *<mptr>* Element

This feature has found some use in such projects as the ECHO DEPository, which uses it to register digital objects at various stages of their ingest into, and dissemination from, a repository;[13] it is also recommended by the PARADIGM project as a method for archiving born-digital content, such as emails.[14]

Nonetheless, its usage remains fairly limited; of all the METS Profiles registered on the central METS repository, for instance, ECHO DEP at the time of writing remains the only project on the Library of Congress's repository of METS Profiles to employ this feature. [15] An important reason for its limited take-up is that its potential for more sophisticated uses than merely populating a division of the structural map is severely limited by its place in the METS schema. The *<mptr>* element can only be used as a direct child of its parent * *: it cannot, for instance, be located in *<par>* or *<seq>* elements to indicate that the objects referenced in its subordinate METS files should be processed in parallel or in sequence (as is required by the different experiment types in figure 1), nor may the contents of these files be processed by the sophisticated partitioning features of the *<area>* element, which allows subsidiary parts of a * * to be addressed directly.

A more sophisticated approach to combining digital object components is to employ Open Archives Initiative Object Reuse and Exchange (OAI-ORE) aggregations,[16] which express more complex relationships at greater levels of granularity than the *<mptr>* method allows.

McDonough's examination of the possibility of aligning the two standards concludes that it is indeed possible, although at the cost of eliminating the METS behavior section and removing much of the flexibility of METS's structural links, both side effects of OAI-ORE's requirement that resource maps must form a connected RDF graph.[17] In addition, converting between METS and OAI-ORE may not be lossless, depending on the design of the METS document.[18]

Neither approach therefore seems ideal for an application of this type, the former because of the limited ways in which the *<mptr>* element can be deployed outside the *<area>* element and its subsidiaries, the latter because of its removal of the functionality of the behavior section, which is essential for the delivery of material such as this.

METS as an Intermediary Schema

An alternative approach adopted here uses the technique of employing METS files as intermediary schemas to act as templates from which METS-encoded packages for delivery can be generated. Intermediary XML schemas are intermediary in the sense that they are designed not to act as final metadata containers for archiving or delivery, but as mediating encoding mechanisms from which data or metadata in these final forms can be generated by XSLT transformations: one example is CERIF4REF, a heavily constrained XML schema used to encode research management information from which metadata in the complex Common European Research Information Format (CERIF) data model can be generated.[19]

The CERIF4REF schema attempts to emulate the architectural processing features of SGML,[20] which are absent from XML; these allowed simpler Document Type Definitions (DTDs) to be compiled for specific applications, which could then be mapped to established, more complex, SGML models. Instead of architectural processing, CERIF4REF uses XSLT to carry out this processing, so allowing the combination of a simpler scheme tailored to the requirements of an application to be combined with the benefits of a more interoperable but highly complex model that is difficult to implement in its standard form.

Instead of using this technique for constraining encoding to a simpler model and generating more complex data structures from this, the intermediary schema technique may be used to define templates, similar to a content model, from which the final METS files to be delivered can be constructed. As is the case with CERIF4REF, XSLT is used for these transformations, and the XSLT files form an integral part of the application. In this way, a series of templates, beginning with highest-level abstractions, are used to generate their more concrete subsidiaries, until a final version used for dissemination is generated. The core of this application is a METS file, which acts as a template for the data delivery requirements for each type of experiment. Figure 2 demonstrates the components necessary for defining these for the *2Grating* experiment subtype detailed previously in figure 1.

```
a
<mets:fileSec>
  <mets:fileGrp USE="experiment-bleach-2Grating">
    <mets:fileGrp ID="twograting-grating1"/>
    <mets:fileGrp ID="twograting-grating2"/>
  </mets:fileGrp>
</mets:fileSec>

b
<mets:structMap ID="top">
  <mets:div>
    <mets:div TYPE="experiment-types">
      <mets:div TYPE="experiment-bleach" ID="experiment-type0001">
        <mets:div TYPE="experiment-bleach-2Grating" ID="experiment-type0001-subtype001">
            <mets:div TYPE="experiment-bleach-2Grating-images" ID="experiment-type0001-
                        subtype001-images">
              <mets:fptr>
                <mets:par>
                  <mets:area FILEID="twograting-grating1"/>
                  <mets:area FILEID="twograting-grating2"/>
                </mets:par>
              </mets:fptr>
            </mets:div>
          </mets:div>
    </mets:div>
</mets:structMap>

c
<mets:behaviorSec>
  <mets:behaviorSec LABEL="XSLT transformations to generate experiment instances">
    <mets:behavior ID=<?"experiment-bleach-2Grating-generate" STRUCTID="top">
      <mets:mechanism LOCTYPE="URL" xmlns:ns2="http://www.w3.org/1999/xlink"
                    ns2:href="experiment-instance-generate.xsl"/>
    </mets:behavior>
  </mets:behaviorSec>

  <mets:behaviorSec LABEL="behaviour for experiment processing">
    <mets:behavior ID="process-non-clem" STRUCTID="experiment-type0001-subtype001">
      <mets:mechanism LOCTYPE="URL" xmlns:ns2="http://www.w3.org/1999/xlink"
                    ns2:href="software/process-non-clem"/>
    </mets:behavior>
  </mets:behaviorSec>
```

Figure 2. Defining an Experiment Subtype in METS

The data model for the delivery of these objects is defined in the *<structMap>* (b): as can be seen here, a series of nested * * elements is used to define the relationship of experiment subtypes to types, and then to define, at the lowest level of this structure, the model for delivering the objects themselves. In this example, two files are to be processed in parallel; these are defined by *<area>* elements within the *<par>* (parallel) element. In a standard METS file, the *FILEID* attribute of *<area>* would reference a *<file>* element within the METS file section (a): in this case, however, they reference empty file group *(<fileGrp>)* elements, which are populated with *<file>* elements when this template undergoes its XSLT transformation.

The final component of this template is the METS behavior section (c), in which the applications required to process the digital objects are defined. Two behavior sections are shown in this example: the first is used to invoke the XSLT transformation by which this METS template file is to be processed, the second to define the software necessary to co-process the two images files for delivery. Both indicate the divisions of the structural map whose components they process by their *STRUCTID* attributes: the first references the map as a whole because it applies to recursively to the METS file itself, the second references the experiment for which it is needed.

When delivering a digital object, it is then necessary to process this template METS file to generate the final version used to encode its metadata in full. The XSLT used to do this co-processes the template and a separate METS file defined for each object containing all of its relevant metadata:

this latter file is used to populate the empty sections of the template, in particular the file section. Figure 3 provides an illustration of the XSLT fragment which carries out this function.

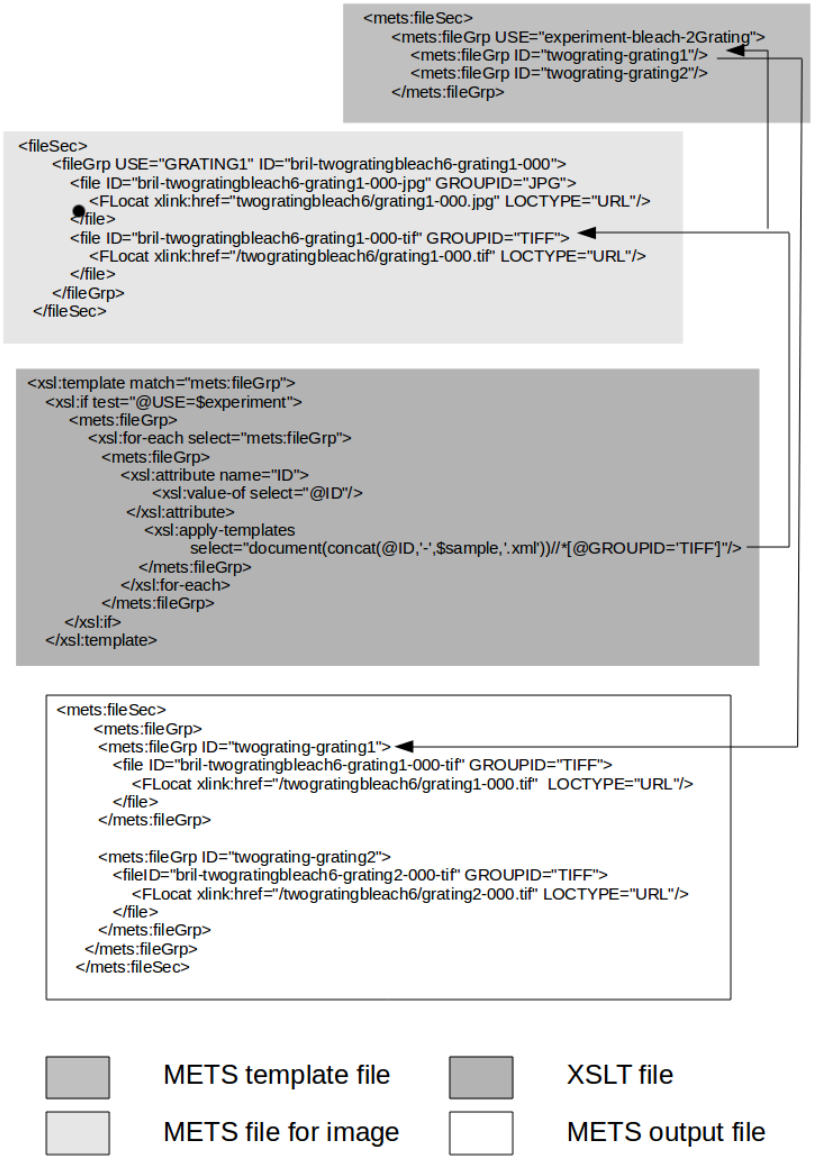

Figure 3.

The XSLT transformation file is evoked with the *sample* parameter, which contains the number of the sample to be rendered: this is used to generate the filename for the *document* function, which selects the relevant METS file containing metadata for the image itself. The *<file>* element within this file, which corresponds to the required image, is then integrated into the relevant *<fileGrp>* element in the template file, populating it with its subcomponents, including the *<FLocat>* element, which contains the location of the file itself.

In the case of the *ACTIN_5* experiment, which generates a video file from a sequence of still images, the processes involved are slightly more complicated. Because the number of still images to be processed will vary for each sample, it is not possible to specify the template for the delivery

version of the sequence explicitly within a *<fileGrp>* element as is done for the other experiments. Instead, it is necessary to define a further METS file (the "sequence file") in which the sequence for a given sample is defined. In this case, the architecture is shown in figure 4.

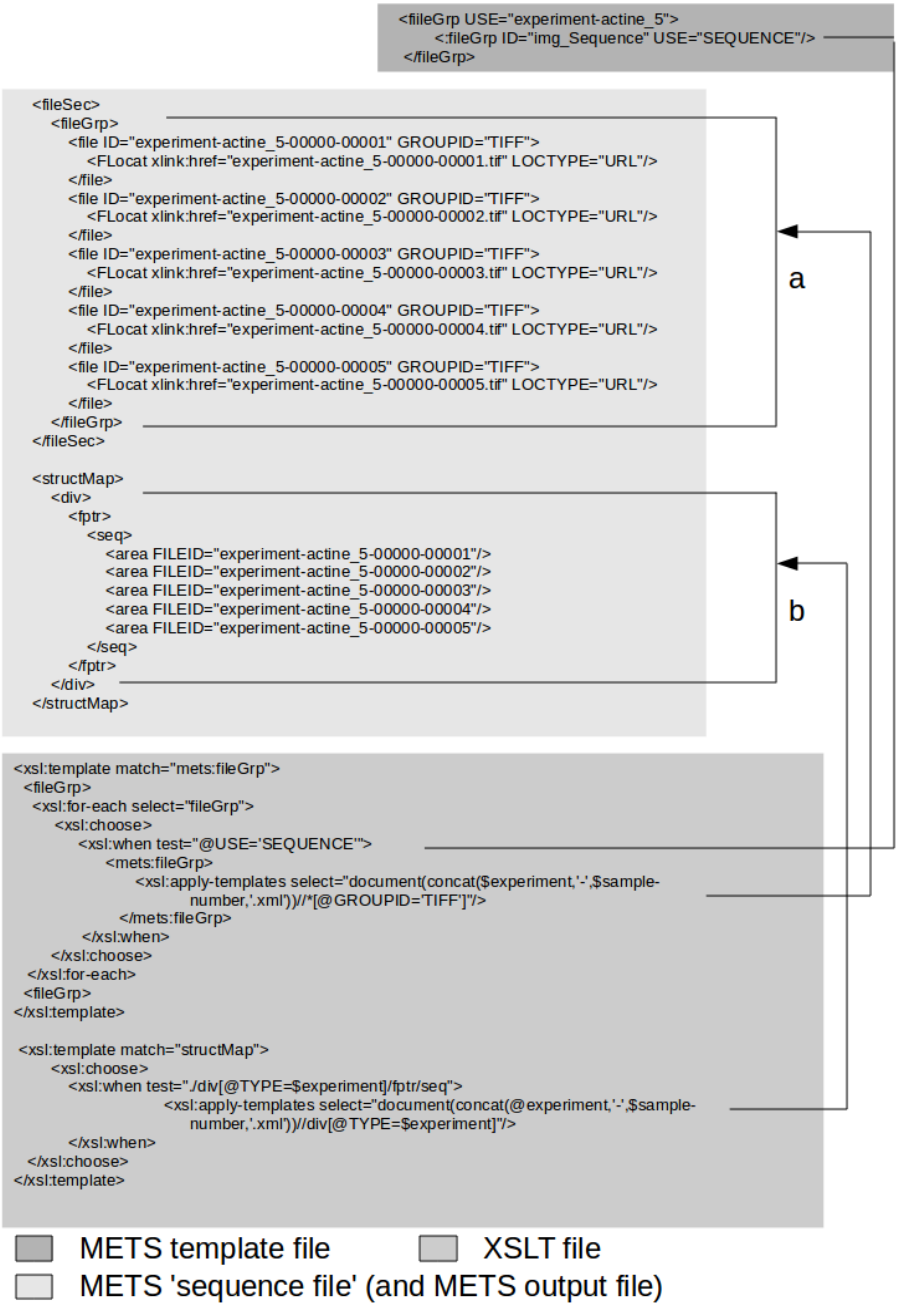

Figure 4. Populating Sequentially Processed File Section with XSLT

In this case the *<fileGrp>* element in the METS template file acts as a placeholder only and does not encode even the skeletal information for the parallel-processed TIFF files in figure 3. Similarly, the structural map * * for this experiment indicates only that this section is a sequence but does not enumerate the files themselves even in template form. Both of these sections are populated when the file is processed by the XSLT transformation to import metadata from the METS "sequence file,"

in which the file inventory (a) and sequential structure (b) for a given sample are listed. The XSLT file populates the file section and structural map directly from this file, replacing the skeletal sections in the template with their counterparts from the sequence file.

Through this relatively simple XSLT transformation, the final delivery version of the METS file is readily generated for either content model. This file can itself then be delivered on the fly (for instance, as a Fedora disseminator); this is done by using a further XSLT file to process the complex digital object components using the mechanism associated with each experiment in the METS behavior section. Given the relatively small size of all of the files involved, this processing can be done more quickly than would be possibly using a fully aggregated METS approach. In the laboratory environment in particular, where the fast rendering and delivery of these images is needed so as not to impede workflows, this has major advantages.

Although the project aimed to examine specifically the use of Fedora for the delivery of this complex material, and so employed FCM as the basis of its metadata strategy, the technique examined in this article proved itself a viable alternative that made much fewer demands on developer time. The small number of XSLT stylesheets required to render and deliver the METS files were written within a few hours: the development time to program the delivery of the RDF-based metadata that formed the FCM required several weeks. Processing XML using XSLT disseminators in Fedora is very fast, and so using this method instead of processing RDF introduces no discernible delays in object delivery.

CONCLUSIONS

This approach to delivering complex content appears to offer the benefits of the alternative approaches outlined above in a simpler manner than either currently allows. It offers much greater flexibility than the METS *<mptr>* element, which can only populate a complete structural map division. When compared to the FCM approach, this strategy, which relies solely on relatively simple XSLT transformations for processing the metadata, requires less developer time but offers a similar degree of flexibility of structure and granularity. It also avoids much of the rigidity of the OAI-ORE approach by not requiring the use of connected RDF graphs, and so frees up the behavior section to define the processing mechanisms needed to deliver these objects.

Using the intermediary schema technique in this way does therefore offers a means of combining the advantages of employing well-defined interoperable metadata schemes and the practicalities of delivering digital content in an efficient manner, which makes limited demands on development. As such, it represents a viable alternative to the previous attempts to handle complex aggregations within METS discussed above.

The adoption of integrated library systems (ILS) became prevalent in the 1980s and 1990s as libraries began or continued to automate their processes. These systems enabled library staff to work, in many cases, more efficiently than they had been in the past. However, these systems were also restrictive—especially as the nature of the work began to change—largely in response to the growth of electronic and digital resources for which they were not intended to manage. New library systems—the second (or next) generation—are needed to effectively manage the processes of acquiring, describing, and making available all library resources. This article examines the state of library systems today and describes the features needed in a next-generation ILS. The authors also examine some of the next-generation ILSs currently in development that purport to fill the changing needs of libraries.

REFERENCES

[1] Library of Congress, "Metadata Encoding and Transmission Standard (METS) Official Web Site," 2011 http://www.loc.gov/standards/mets (accessed August 1, 2011).

[2] Organisation Internationale de Normalisation, "ISO/IEC JTC1/SC29/WG11: Coding of Moving Pictures and Audio," 2002, http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm (accessed August 1, 2011).

[3] Fedora Commons, "The Fedora Content Model Architecture (CMA)," 2007, http://fedora-commons.org/documentation/3.0b1/userdocs/digitalobjects/cmda.html (accessed December 9, 2011).

[4] Carl Lagoze et al., "Fedora: An Architecture for Complex Objects and their Relationships," *International Journal on Digital Libraries* 6, no. 2 (2005): 130.

[5] Ibid., 127.

[6] Ibid., 135.

[7] Ibid.

[8] Rishi Sharma, *Fedora Interoperability Review* (London: Centre for e-Research, 2007), http://wwwcache1.kcl.ac.uk/content/1/c6/04/55/46/fedora-report-v1.pdf.3 (accessed August 1, 2011).

[9] Richard Gartner, "Intermediary Schemas for Complex XML Publications: An Example from Research Information Management," *Journal of Digital Information* 12, no. 3 (2011), https://journals.tdl.org/jodi/article/view/2069 (accessed August 1, 2011).

[10] Centre for e-Research, "BRIL," n.d., http://bril.cerch.kcl.ac.uk (accessed August 1, 2011).

[11] S. J. DeRose et al., "What is Text, Really," *Journal of Computing in Higher Education* 1, no. 2 (1990): 6.

[12] Digital Library Federation, "<METS>: Metadata Encoding And Transmission Standard: Primer And Reference Manual," Digital Library Federation, 2010, www.loc.gov/standards/mets/METSPrimerRevised.pdf, 77 (accessed August 1, 2011).

[13] Bill Ingram, "ECHO Dep METS Profile for Master METS Documents," n.d., http://dli.grainger.uiuc.edu/echodep/METS/DRAFTS/MasterMETSProfile.xml (accessed August 1, 2011).

[14] Susan Thomas, "Using METS for the Preservation and Dissemination of Digital Archives," n.d., www.paradigm.ac.uk/workbook/metadata/mets-altstruct.html (accessed August 1, 2011).

[15] Library of Congress. "METS Profiles: Metadata Encoding and Transmission Standard (METS)

OfficialWeb Site", 2011. http://www.loc.gov/standards/mets/mets-profiles.html (accessed December 6, 2011).

[16] Open Archives Initiative, "Open Archives Initiative Protocol—Object Exchange and Reuse," n.d., www.openarchives.org/ore (accessed December 12, 2011).

[17] Jerome McDonough, "Aligning METS with the OAI-ORE Data =Mmodel," *JCDL '09 Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital libraries* (New York: Association for Computing Machinery, 2009): 328.

[18] Ibid., 329.

[19] Gartner, "Intermediary Schemas."

[20] Gary Simons, "Using Architectural Processing to Derive Small, Problem-Specific XML Applications from Large, Widely-Used SGML Applications," *Summer Institute of Linguistics Electronic Working Papers* (Chicago: Summer Institute of Linguistics, 1998), www.silinternational.org/silewp/1998/006/SILEWP1998-006.html (accessed August 1, 2011).