PREMIS Editorial Committee Conference Call Notes

19 July 2007

In attendance: Steve Bordwell, Olaf Brandt, Rebecca Guenther, Bill Leonard, Zhiwu Xie, Gerard Clifton (notes).
Apologies:, Priscilla Caplan, Brian Lavoie, Yaniv Levi, Rory McLeod.

1. Tutorial materials

The question was raised whether tutorial materials should be made more publicly available. The materials are currently available on the PIG wiki, but it was suggested that materials be made available from the LC PREMIS page. It was agreed that only the most recent tutorial materials (e.g. the tutorials held at LC) should be made available on the 'Resources' page of the Maintenance Activity site. All the materials associated with the tutorial should be made available, not just the presentation slides. If there are significant changes to materials for later tutorials, the updated materials should replace existing materials as most current.

ACTION: [Rebecca Guenther ?] to post the most recent tutorial materials to the Maintenance Activity 'Resources' page

2. Group discussion of DD changes

Change 22 – Disambiguating source and outcome objects associated with an event

Issue: "Another issue with Events and linkingObjectIdentifier is that there is no way of saying (if applicable) which was the "source" object and which was the "output" object other than by referring back to one of the objects to find its relationship to the other object, and again there is the potential for inconsistency… The issue is that linkingObjectIdentifier does not allow you to distinguish which object is the "source" of the event, and which object is the "outcome" of the event. To obtain this information, you would have to take the extra step of going to the metadata associated with the objects and checking the relationship documented between them."

Proposal: Brian proposed solutions to this issue (see http://listserv.loc.gov/cgi-bin/wa?A2=ind0707&L=premis-ec&T=0&P=1748):

1. Add an optional sub-element called linkingObjectRole to linkingObjectIdentifier in the Event entity. This could be populated with terms from a controlled vocabulary, like "source" or "outcome". This semantic unit would serve the same function as linkingAgentRole (an existing sub-element of linkingAgentIdentifier in the Event entity): i.e., to provide further disambiguation beyond identification.

OR

2. Note source and outcome objects in eventOutcomeDetail.

Discussion: There was discussion about whether it was necessary to identify source and output objects, and whether this was core metadata: whether it is something which should be left to the repositories to decide or should be made explicit. Steve noted that in the NAS repository, an originating ID is used to identify a source object, and that this information is indeed core information in the NAS repository context.

Decision: Use of a new semantic unit 'linkingObjectRole' is the preferred option, to be coupled with 'linkingObjectIdentifier'. 'linkingObjectRole' should be optional: it won't apply in all cases, and may be required only to disambiguate object roles.

Change 12 – creatingApplication and the onion model

Issue: "…in the onion model, you would repeat the objectCharacteristics block for each compositionLevel. So, if PDF file A is gzipped, I have one objectCharacteristics block for A.pdf (compositionLevel 0) and one for A.pdf.gz (compositionLevel 1). Each of these has a separate and distinct creatingApplication. I can record both creating applications because creatingApplication is repeatable, but I can't link the creatingApplication element to the appropriate objectCharacteristics/compositionLevel… " (see http://listserv.loc.gov/cgi-bin/wa?A2=ind0602&L=pig&T=0&P=3763)

Proposal: Gerard provided two proposals (see http://pec.lib.uchicago.edu:8888/pec/22):

1. Add a new semantic unit, 'relatedCompositionLevel', under the creatingApplication container, the value of which should correspond, on a per object basis, to the appropriate compositionLevel recorded under objectCharacteristics.

OR

2. Move creatingApplication back under objectCharacteristics, and allow objectCharacteristics (or an equivalent for Representations) to apply at the Representation level.

Discussion: Discussion included a number of points:

- Multiple bundled objects would need to be described as separate objects (see DD pg 4-4). The relationship indicated by corresponding values in compositionLevel (0,1,2) and relatedCompositionLevel could only apply on a per object basis, to avoid confusion for multiple '0' values for instance, but this may be a source of confusion anyway.

- A relationship between a creatingApplication applied at the Representation level and a compositionLevel cannot be asserted, as compositionLevel (and objectCharacteristics) cannot be applied to the object at the Representation level.
- It is simpler to move creatingApplication under objectCharacteristics; the relationship between compositionLevel and creatingApplication is then clear.
- Potential uses for creatingApplication as part of events recorded in <digiProv> sections in METS could be handled by other semantic units (e.g. linkingAgentIdentifier in an Event, which could refer to a software Agent).

Decision: The agreed preference is to move creatingApplication back under objectCharacteristics (option 2) and then solve the issue of allowing objectCharacteristics to apply at the Representation level (which has been discussed previously and is under review).

Change 7 – Storage

Issue: Interpretation of 'storage' and 'contentLocation' in regard to files: is it the physical location of the file, or how the file can be located (not the same thing) (see http://listserv.loc.gov/cgi-bin/wa?A2=ind0606&L=pig&T=0&P=1913)

Proposal: Zhiwu provided a proposal (see http://pec.lib.uchicago.edu:8888/pec/23):

- contentLocationType as a contained element of contentLocation should be information specific and directly about the storage system. We must not assume any resolution schema except for the one used natively by the storage system. Information that involves a resolution system should not be used.

Discussion: There was debate over this topic, with the following points raised:

- As these elements are for recording storage location, they should refer only to native storage locations, and not use an identifier that requires resolution by another service which requires an infrastructure overhead. The native mechanism used by the storage system for recording location should be used.
- It was noted that storage locations, even absolute file system locations, are abstractions anyway – file references which are further resolved by native systems. A file system is just a service provided by a system for referencing files. There may not be a discrete storage location, nor a unique native reference to it – Unix can offer multiple mount points for the same file. Some storage systems are opaque and locations are not explicit.
- There was a desire to keep this element open to allow Web references.
- This question may be implementation specific. It should be up to repositories to determine the level of granularity they require.

Decision: Outcome of discussion was:

- Repositories should be able to record values appropriate to their implementation, and at whatever level of granularity is required.
- In the DD semantic unit description, the 'resolvable' identifiers should remain as valid values; however, file system examples should also be included, in order to make the range of valid choices for location identifier types and granularity more apparent.

ACTION: Zhiwu Xie to enhance the definition and examples for contentLocationType and contentLocationValue.

Change 8 – storageContainer optional for bitstreams

Issue: storageContainer should be optional for bit streams. Should storage not be applicable to bit streams and a new element defined for recording the location of a bit stream? (see http://listserv.loc.gov/cgi-bin/wa?A2=ind0602&L=pig&T=0&P=1084)

Proposal: Zhiwu provided a proposal (see http://pec.lib.uchicago.edu:8888/pec/24)

- It is not clear why storageMedium does not apply to Bitstream. People may be able to infer this information from the file that contains the Bitstream, however contentLocation is optional, so it's not always clear in which file a bitstream is contained. It's better to explicitly record this information.

Discussion: Discussion on this topic raised the following points:

- Location of a bitstream may be a byte offset within a file. If we know the offset location, we may want to record it.
- Currently, we may only know about bitstreams in terms of their relationship to a file. StorageMedium characteristics for a bitstream are implied by the relationship between a file and a bitstream. However, the recording of the relationship between files and bitstreams is optional, so if the relationship is not recorded, any characteristics implicitly inherited by the bitstream from the file would be lost. It may be better to explicitly record these characteristics against the bitstream.
- It may be simpler to requote the storageMedium and contentLocation at the Bistream level, rather than require the relationship to be recorded.
- In the usage notes we should assert that a repository should know about the relationship between a file and a bitstream (but we should not say how this should be implemented or recorded).
- There may be cases where the location of the bitstream is not known.

- In the proposal, storageMedium is proposed as mandatory for Bitstream. However, contentLocation (as a container) is optional, so if the contentLocation is not recorded it may not be useful to record storageMedium for Bitstream.

Decision:

- Make storageMedium applicable at the Bitstream level, but optional.

6. Other business – Olaf Brandt will be regularly calling in from the Koninklijke Bibliotheek from August 1, and so will require the dial-in number for the Netherlands.
ACTION: [Rebecca Guenther ?] to advise Olaf of the conference dial-in number for the Netherlands.

Next call: 2 August 2007.