Towards seamless integration of Digital Archives with source systems (Part 2)

Alan Gairey[1], Kevin O'Farrelly[1] , Robert Sharpe [1]

[1] Tessella Ltd, 26 The Quadrant, Abingdon Science Park, Abingdon OX14 3YS UK

Abstract

When integrating record sources (e.g., EDRM systems, web archiving software, ad-hoc submissions etc.) to a digital archiving system it is common to have to translate from a variety of source schemas into an archival metadata schema. This can lead to some loss of information owing to imprecise mappings. In addition, both the source and archival schema are subject to change over time meaning that maintaining the necessary transforms is also costly. It is possible to try to reduce the problem by mandating some standards but, given the long timescales for archiving, this can only reduce (see Part 1) but not eliminate these problems. Inevitably, the archival system either still needs to cope with heterogeneous metadata schemas or needs to perform even more (potentially lossy) metadata transforms.

In Part 2 we present an alternative approach that has simplified the integration between source systems and the long-term archive in a number of institutions by allowing each type of record to use the original descriptive schema used by its source (i.e., use metadata fields appropriate for that type). Appropriate technologies are used to ensure that it is still possible to view, edit (where appropriate) and search within each metadata field. Thus, the archive's internal metadata schema need only be responsible for maintaining structural, technical and preservation metadata. Hence, the archive allows all the functionality that would be expected if a fixed descriptive schema were used without the need for complex mappings.

Introduction

Digital content of archival value needs to be ingested into a digital repository to enable it to be maintained over long timescales. The characteristics of such a digital repository should comply with the Open Archival Information Systems (OAIS) reference model [1]. This defines both a functional model and an information model with which such repositories should comply.

One of the six functional entities defined by OAIS is Ingest. This function enables digital content to be taken from external (source) systems and deposited into the repository. In addition to content, metadata from the source system needs to be transferred into the archival system ideally with zero loss of information and yet in such a way that the metadata is still readily understood outside of its source system.

The OAIS information model helps analyse this issue by defining a number of entities that need to be preserved and thus must be identified as part of the ingest process. The conceptual entity of real interest is called an Information Object. This is equivalent to an archival record. As with records, Information Objects can be recursive so high-level Information Objects contain subsidiary Information Objects.

Digital Information Objects contain both the Digital Object (usually a collection of files that need to be ingested) and Representation Information. For digital records, Representation Information is defined as the information needed to map the Digital Object into "more meaningful concepts". In practice this means that it consists of:

- Semantic Information that describes the Information Object in terms that are not dependent on the digital technology utilised by the Digital Object. Ideally this information should not change if the technology used by the Digital Object needs to be refreshed (e.g. by migrating to a new format).
- Structure Information that describes how the Information Object is manifested by the Digital Object (e.g., the list of files and relative paths of these files).
- Technical metadata about the Digital Object that enables its preservation (e.g., the format of each file in the Digital Object). This usually references an external source with more information that is needed to complete this description (such as a format registry like PRONOM [2]).

Of these, the structural information is usually derived directly from the Digital Object. One option is to treat a rich structure of files as a single Information Object. Alternatively, it is possible to create a conceptual hierarchy of Information Objects based on, say, the folder structure of the high-level Digital Object with each folder in this structure being used to posit the existence of a subsidiary Information Object. The exact method used is dependent on the circumstances.

In the main, Technical metadata is also directly derivable from the Digital Object. This typically involves physical characterisation [3, 4] through a process of format identification, format validation, property extraction and detection and extraction of embedded objects. These embedded objects are then characterised in turn, continuing recursively until no further recursion is possible pragmatically. In addition it can include conceptual characterisation [4] where the existence of a network of technology-independent components in need of preservation is detected together with their essential characteristics that should be maintained regardless of any actions (e.g., format migrations) that might be needed because of technology obsolescence.

However, semantic information is not usually derivable from the Digital Object. Fortunately in most archival scenarios the Information Object has already been stored in some previous system (e.g., an electronic document records management system, EDRMS). This means that semantic information will often already exist in some form.

This information can be expressed in a number of ways. In some cases this might be expressed in one of a number of standard schemas such as Dublin Core [5], Encoded Archival Description [6], Metadata Object Description Schema [7] etc. In many cases this will be held in a proprietary from (e.g., in database tables) determined by the EDRMS or other system. In some of the latter cases it might still be possible to export the information to a more standard form but even here there is the possibility of information loss in the transformation since it is not always easy to map information from one form to another.

One way to get around this problem is to standardise the metadata throughout the source organisations. This can work if all organisations are under a common jurisdiction (e.g., a single country's government departments and agencies which will be archiving their content to a national archive). This approach certainly reduces the variation and is discussed in the companion paper preceding this. However, this will not work when an archive receives

content from multiple jurisdictions and, anyway, such standards are always likely to be refined. Hence, whilst this is a sensible approach it does not mean that a long-term repository can ignore the inevitable need to ingest multiple variants of such a standard schema.

In this paper we report on an alternative approach where the repository is designed up-front to allow for variation in semantic information schemas. The difficulty with creating such a system is that a repository needs to provide functionality that interacts with individual fields in the metadata to allow fielded metadata viewing and editing, fielded searching and the ability to perform other functions based on the values of specific fields. However, by only making the restriction that the metadata must be expressed in an XML document (in *any* schema) it is possible for a repository still to offer all of these features whilst allowing heterogeneous schemas to be used.

This approach has been utilised in practice in Tessella's Safety Deposit Box (SDB) digital preservation system [8] including the cloud-based offering Preservica [9] and is in use by a range of organisations to store descriptive metadata for a wide variety of descriptive sources.

Types of Metadata

To alleviate the issue just described with heterogeneous sources and heterogeneous metadata schemas a repository needs to makes a clear distinction between:

- Structural Metadata. This is needed to:
    - Identify the existence of each Information Object.
    - Identify the relationships between Information Objects (both hierarchical, e.g., parent and child relationships, and cross-link relationships).
    - Identify the digital files that manifest the Information Object in a particular technology.
    - Identify the multiple manifestations of that Information Object that represent the object in different combinations of technology and/or for different purposes (e.g., a master preservation copy and a presentation copy, such as a lower resolution representation of an image).
    - Identify relationships between multiple manifestations to record information on the transformations that have occurred (assuming they occur within the system).
- Technical metadata. This is needed to:
    - Identify the format of each digital file and whether the file successfully validates against the formal specification of that format.
    - Identify key properties of each file which can be used to identify potential preservation issues and/or to determine significant properties that must be preserved (i.e., remain invariant within some tolerance) in future preservation actions.
    - Identify embedded objects within each file that also need to be preserved and also might lead to a potential preservation issue.
    - Identify embedded objects recursively within other embedded objects.
    - Identify the existence of a network of technology-independent "components" and their properties that need to be preserved even if the technology used to manifest the Information Object changes radically (e.g., a m-to-n file migration so file structure is not preserved).
- Semantic or descriptive metadata. This is needed to enable human beings to interpret the Information Object correctly.

Both structural and technical metadata are vital to perform automated long-term preservation of the Information Objects in the repository. As such they need to be readily machine-interpretable by the preservation system. Hence, SDB uses its own metadata schema (XIP) to describe this metadata. The technical metadata part of this contains everything that is contained in PREMIS [10] plus some SDB-specific extensions [4]. The structural metadata allows efficient storage of multiple manifestations (e.g., not requiring detailed file information to be repeated if a file is present in more than one manifestation). It is possible to export XIP to standard schemas such as METS [11] (with embedded PREMIS metadata).

However, descriptive metadata does not drive such automated functionality and so there is no reason for the repository to impose its own structure on this metadata. Hence, SDB simply allows descriptive metadata to be embedded using any descriptive metadata schema. The only restriction is that it is expressed in XML. Also, since sometimes objects may need to be described in different ways for different purposes, SDB allows multiple schemas to be used to describe each entity. Clearly it is a good idea to choose multiple schemas with minimal repetition so there is little overlap but this additional flexibility can be useful, e.g., if the entity already has a description in multiple systems that might need to be combined.

Despite this flexibility, the descriptive metadata is involved in a variety of functions in the repository namely:

- During ingest, ensuring that the metadata is extracted from the source and safely stored in the repository.
- Allowing users to view the fielded metadata.
- Allowing authorised users to edit any field in the metadata.
- Allowing users to search within any field of the metadata.
- Allow other functionality to be dependent on the value of any combination of fields within the metadata.

Hence, the rest of this paper describes how these functions are all achieved without imposing restrictions on the descriptive metadata.

Ingest

OAIS defines the package that is used to start ingest as the Submission Information Package (SIP). A SIP needs to contain not just the digital files but all of the metadata needed for archiving that cannot be derived from the content automatically. This includes descriptive metadata. Hence, this descriptive metadata needs to be extracted from its source and associated correctly with the relevant Information Object.

The exact way that this ingest occurs will vary from source to source. For example, if there is a direct connection between the archival repository and the source system (for example, via an API) then metadata extraction and association will usually occur at the same time (e.g., the source system is queried to determine which objects need archiving and these objects are extracted with their metadata in one operation). In this way a SIP can be built.

An alternative (in the absence of an API on the source system) is to export content from the source in some appropriate structure (e.g., a folder structure that reflects the Information Object hierarchy) and include metadata fragments with each relevant extracted entity. Then the ingest process can correctly and automatically interpret which digital files are content and which contain descriptive metadata and thus build a SIP from the result of the export.

In the event that some content may not be held in a recognised source (e.g., archival material donated by the public), it is possible to follow a similar approach to creating a SIP, e.g., creating Information Objects based on the physical hierarchy as if it were an export from a source system. Since there is no requirement to use a particular descriptive metadata schema, it is also possible to create a SIP quickly by adding only whatever descriptive metadata is available at whatever level it is available (e.g., by adding some general metadata just to the top-level Information Object).

Other material might be extracted from non-controlled sources (e.g., when crawling web sites) and again, as much metadata as is possible can be added to each entity, but equally if no such metadata is available the ingest is not held up by the need for a human being to make some up. This is a good example of the flexibility of the approach since information can be added here about, for example, the original URL and crawling information without the need to shoehorn the information into fields in standard schemas that were not necessarily designed to hold that information.

In short, each Information Object in a SIP can utilise the most appropriate metadata schema (or combination of metadata schemas) to describe itself. Hence, regardless, of the source, this allows a SIP to be created with no loss of information. This is the key point. Alternative approaches involve transformation of information and this can lead to information loss. Of course, the approach outlined above does not prevent transformation of metadata: it just does not *require* it. This means there is no need to wait for the time needed to perfect the metadata before ingesting content into a digital repository. Instead it can be ingested quickly to ensure it is kept safely and transformation (if needed at all) can take place at a later date (when it will take place within a fully audited environment, which means if it is discovered later that this has caused some information loss, the loss can be reversed).

Viewing and editing metadata

Clearly metadata embedded in an XML schema is held in the repository but it is not in a form that is easily visible to human beings. To get around this, metadata can be transformed from XML into HTML through the use of an Extensible Stylesheet Language Transformation (XSLT).

In a similar way, to allow authorised users to edit the metadata the metadata can be transformed into a different HTML representation using an alternative Extensible Stylesheet Language Transformation. This editing is always verified to ensure that the output (a new XML document) remains compliant with the appropriate XML schema. In addition, further verification can be configured as required.

It is possible to add additional Extensible Stylesheet Language Transformations to allow for further transformations between schemas. This might be useful if some rationalisation of schemas is attempted after ingest or during export of an entity for a particular usage scenario.

One of the advantages of this approach is that the full description of each entity is present in an appropriate fragment of an XML document. Hence, following an edit, all that is needed is that a new fragment is created and to ensure that the entity being edited uses this fragment from now on. This means that systems can maintain a history of all fragments automatically and provide an audit trail of all changes that have occurred. It can then allow each individual entity to be rolled back to any previous state, should that be required.

Fielded metadata searching

Another potential issue with allowing any metadata schema to be used is to ensure that users can still perform fielded search, even though the repository will not know up-front which fields will be present.

In early versions of SDB this problem was solved by allowing XPath expressions to be used to describe each field. However, this was a time-consuming exercise and thus created a considerable overhead in adding new schemas. Hence, this system has now been replaced by including the open-source search-server Apache Lucene [12]. This automatically indexes all metadata in any schema, which allows fielded metadata searching to be enabled with minimal configuration effort. It is still possible to enhance the default behaviour if the XML schema is non-optimal (e.g., by allowing an XML text field that actually holds a date to be treated as such and hence allow inequality based searching).

Other functionality

SDB is a workflow-based system so any functions that need to be performed (e.g., ingest, storage, access, data management and preservation) are performed through a series of workflow steps. There are a series of core workflow steps that combine to produce a set of generic workflows that can be used to perform common features (e.g., validated migrations of content based off a format considered to be at risk).

The system is very flexible and allows new workflow steps to be combined with the existing workflows steps to produce new workflows that perform different functions. These workflow steps can utilise an API to look up any archival information (including fields in descriptive schemas) and use this information to determine which actions to perform. Hence, the approach of allowing heterogeneous schemas does not preclude performing actions that vary depending on the values of any field in any type of record.

Administration

SDB allows administrators to keep track of the schemas allowed in the system by uploading schemas into the system. In addition, the viewing, editing and other transforms can also be uploaded. This means that the administrative process of adding new descriptive schemas is simple and ensures that the repository is self-describing in the sense that the information required to interpret the descriptive metadata is held within the repository itself.

Conclusions

This paper has described how a digital repository can allow heterogeneous descriptive metadata schemas to be used whilst still allowing the features that normally drive systems to choose a single fixed schema: in particular, viewing and editing of every metadata field and fielded searching. This approach dramatically reduces the overhead required to ingest digital material from any new source and allows many organisations to use the same software without having to agree to use a single shared metadata schema. In addition, since it does not require metadata transformations to occur, it reduces the possibility of information being lost as a consequence of ingesting it into an archive. It still allows transformation to occur at a latter date but does so in an audited (and reversible) environment, which means any

information loss need not be permanent. Even in scenarios where a single schema is likely to be used it allows for future proofing, since new versions of that schema can be put into operation easily without requiring a complex upgrade and data porting exercise.

This approach has been demonstrated to work in reality through its implementation in Tessella's SDB and Preservica products and as such is in operational use in a large number of archives and libraries throughout the world who are the forefront of digital preservation.

References

[1] ISO 14721:2003 (http://www.iso.org/iso/catalogue_detail.htm?csnumber=24683)
[2] http://www.nationalarchives.gov.uk/pronom/
[3] Adrian Brown (2007), Developing Practical Approaches to Active Preservation. The International Journal of Digital Curation, Issue 1, Volume 2
[4] Robert Sharpe (2010) Active Preservation of web sites, Proceedings of International Web Archiving Workshop IWAW 2010
[5] http://dublincore.org/
[6] http://www.loc.gov/ead/
[7] http://www.loc.gov/standards/mods/
[8] http://www.digital-preservation.com/solution/safety-deposit-box
[9] http://www.preservica.com
[10] http://www.loc.gov/standards/premis/
[11] http://www.loc.gov/standards/mets
[12] http://lucene.apache.org/solr