A METS BASED INFORMATION PACKAGE FOR LONG TERM ACCESSIBILITY OF WEB ARCHIVES

Markus Enders
The British Library
96 Euston Road
London, NW1 2DB

ABSTRACT

The British Library's web archive comprises several terabyte of harvested websites. Like other content streams this data should be ingested into the library's central preservation repository. The repository requires a standardized Submission- and Archival Information Package.

Harvested Websites are stored in Archival Information Packages (AIP). Each AIP is described by a METS file. Operational metadata for resource discovery as well as archival metadata are normalized and embedded in the METS descriptor using common metadata profiles such as PREMIS and MODS.

The British Library's METS profile for web archiving considers dissemination and preservation use cases ensuring the authenticity of data. The underlying complex content model disaggregates websites into web pages, associated objects and their actual digital manifestations. The additional abstract layer ensures accessibility over the long term and the ability to carry out preservation actions such as migrations. The library wide preservation policies and principles become applicable to web content as well.

1. INTRODUCTION

The web has become one of the primary information resources. Its information is read by the general public, cited by researchers and re-used by bloggers and commercial publishers. But information on the web is transient. Unlike printed books or journals information can easily be modified or deleted from electronic systems.
Since the mid 90ies when national libraries and the Internet Archive started archiving the web, the importance and the awareness for preserving the information published on web raised.
Today various web archives exists providing access to millions of web pages. The British Library's web archive contains more than 23516 instances of websites comprising of 5.5. TB of (compressed) data. As the size and use of the web archive grows, it becomes important integrating the web archive with the library's preservation system to ensure its long term availability. Therefore the library's preservation policies, supported preservation formats and preservation use cases must be considered.

2. WEB ARCHIVING

The British Library has set up a complex technical infrastructure for collecting, storing and providing access to web sites. A set of tools such as the Heritrix crawler, the Wayback Machine and the Web Curator Toolkit are used. These tools implement common interfaces and use the same file formats for exchanging data.
Legacy data in the archive had been harvested using the PANDAS system. It provides a different infrastructure for harvesting, storing and managing the web archive. As a result the file formats and interfaces are different.
From the long term preservation perspective it is not useful supporting two different formats for the same purpose. The SIPs and AIPs for the Digital Library System are standardized. The supported file formats are consolidated and unified.

2.1. Tools

2.1.1. PANDAS

As a member of the UK Web Archiving Consortium project, the British Library started very early to collect and store web pages. The only tool available at this time was the PANDAS system[1]. PANDAS provides all the required functionality for selecting, harvesting, managing and providing access to websites. As it was one of the first tools, it had scalability problems managing a huge number of concurrent crawls initiated by various curators. The output from PANDAS is very simple: Harvested bytestreams are stored in a directory structure in the local file system. The British Library did not capture comprehensive descriptive metadata nor logfiles.

2.1.2. Heritrix and the Web Curator Toolkit

The current solution for harvesting webpages consist of three components which provide an end-to-end process for harvesting, managing and disseminating archived webpages. It is much more scaleable than the format PANDAS solution.
The internet archive's Heritrix[2] crawler is used to harvest webpages. It starts with a one or more UDLS (so called seed-URLs), analyzes the received bytestream and extracts further URLs from HTML pages. Comprehensive configuration options as pattern matching for URLs, support of robots.txt and counting the crawl depth allows to restrict a single crawl to a specific area of the web. For the selective harvesting the crawl is usually restricted to a single web site.

[1] http://pandora.nla.gov.au/pandas.html
[2] http://crawler.archive.org

The Web Curator Tool (WCT) is used by curators for managing the content and initiating crawls. The WCT allows the curator to set configuration options for the crawler and to schedule crawls for each target. A target is any portion of the web which the curator regards as important to collect and archive. Each target has at least one instance. This target instance is a snaphot of target at a particular point in time. Every time Heritrix is harvesting a target, a new instance is generated.

The Access Tool provides end user access to the harvested content. It uses the metadata which had been captured and generated by the Web Curator Tool as well as the content data which had been harvested by Heritrix. It integrates the open source version of the Wayback machine to access the individual bytestreams which had been harvested by Heritrix.

2.2. Data model of the Web Curator Toolkit

The OAIS model defines three different information packages for submission, archiving and dissemination. They provide the data which is needed to support the appropriate functionality. Information packages are an abstract concept which encompasses all the data being needed for a well defined set of functions. The information packages of current web archiving tools are focused on supporting submission and dissemination. The data structures and information are stored in a convenient way for the Access Tool and the Wayback Machine to disseminate the data.
Information packages can be split over various files, database records etc.

The web archiving toolset uses a so called ARC[1] container for storing the content data. It contains the actual bytestreams being returned as a result of every successful http-request. They are enriched with basic technical (size of the bytestream, mime type) and provenance (date and time of harvest) data. The ARC files are created by Heritrix. Every ARC file is accompanied by an index file. It allows non-sequential access, as it records the location of each URL within the ARC container.
Descriptive and rights metadata are not stored in the ARC container, but in a relational database. The information in the database is captured by curators using the Web Curator Tool.
The information in the The Heritrix crawler creates a number of different logfiles for each crawl. These logfiles contain provenance information about the harvested and stored bytestreams as well as those requests which failed. Failed requests may or may not return a bytestream. In case of http-errors the web servers are usually returns a bytestream and an appropriate error code in the http-header. This bytestream is stored in the ARC container. Other errors such as runtime errors of the software may not return a bytestream. The only evidence of such a request is recorded in the logfile. The logfile enables the curator to retrace the crawler's path through a website and discover the reason if the bytestream for some URLs is not available in the archive.

For provenance purposes the crawler's configuration is also stored. It stored the schedule for regular crawls which is set by the curators using the Web curator Toolkit. They are also responsible for configuring the crawler regarding the crawl depth and URL-patterns. These settings define the conditions for Heritrix to stop following hyperlinks. Besides this process related information, the Web Curator Tool allows the curator to capture descriptive metadata for each target. The metadata is used for resource discovery purposes.

According to the OAIS model both tools – PANDAS as well as WCT/Heritrix - are creating information packages which are used for submission (SIP) and dissemination (DIP). Their content model is defined by the technology being used and optimized for collecting and providing access to webpages. Long term preservation requirements had not been considered. Both SIPs support different standards and are structured differently. As a consequence the integration with other systems, including the library's long term preservation repository is very poor. The systems being used for web archiving are using their own technical infrastructure.

For preserving web content in the long term, the content model must be harmonized. The archival store can only support a single format for the Archival Information Package. This format must be based on common standards and consider long term preservation requirements. Besides the operational metadata being embedded in the SIP/DIP, additional archival metadata must be generated and stored.

3. PRESERVATION REQUIREMENTS

In the long term it will be difficult for the library to run and maintain different systems for storing, managing and preserving information. A library's Digital Library System (DLS) is responsible for storing Archival Information Packages from various sources and various content streams. As a consequence web content must to be ingested into the library's archival store as well. The same common standards and policies must be used.

The Digital Library System serializes metadata as well as content information as files in its internal file system. While the content is stored in so called container files, the descriptor of each package is serialized using the Metadata Encoding and Transmission Standard (METS)[2]. METS is a framework for describing a digital resource and all components of it. In this case it describes a single instance of a website harvested at a particular point in time (target instance). Every instance is stored in a separate information package.
The METS description of the resource comprises descriptive, technical and preservation metadata as well as

[1] http://www.archive.org/web/researcher/ArcFileFormat.php

[2] http://www.loc.gov/standards/mets

the internal structure of the resource. The internal structure defines all the objects the resource consists of: abstract entities such as website and webpages, container files and bytestreams as well as their relationships.. METS uses so called extension schemas such as PREMIS, MODS, Dublin Core to store descriptive and preservation metadata Though a different schema is used, this metadata is part of the METS file.

The library's Digital Library Systems (DLS) stores the METS file and content files in its internal file store. The file store hold three distributed copies of every file. The actual content files are bundled in container files which are similar to the Heritrix output. Instead of ARC the standardized WARC format is used for the container files. All ARC containers had to be migrated to WARC prior to ingest as the DLS' ingest interface accepts only WARC containers with an appropriate METS descriptor.

Storing and preserving bytestreams is just one prerequisite for ensuring the accessibility of information in the future. File Formats, transport protocols and the supported software will change over the next decades. It is uncertain if web browsers and underlying HTML pages will still be the tools and formats of choice and how future software tools will be able to render today's web pages. Preservation actions such as emulation and migration will ensure that the information can still be rendered. Though the web page's manifestation (the bytestream) may change the appropriate documentation ensures information's authenticity.

Preservation metadata ensures the authenticity. It makes "digital objects self-documenting over time"[1] and is an important part of each Archival Information Package (AIP) record. It includes technical details on structure and format of a bytestream as well as the history of all actions taken to maintain the bytestream's information. It is part of the digital provenance metadata for each bytestream which is partly captured when it is harvested. Additional actions may occur prior to ingest into the repository: virus checks, format migrations or other transformations. These actions are properly documented in the preservation metadata record. In case preservation actions result in new bytestreams, the relationship between the old and new bytestream is recorded.

Though the AIP's data structure focuses on supporting preservation, it must consider dissemination use cases as well. Some functions of the access system rely on metadata which needs to be provided the AIP. According to the OAIS model, the Dissemination Information Package (DIP) is derived from the AIP.

[1] PREMIS Data Dictionary for Preservation Metadata, version 2.0, March 2008, http://www.loc.gov/standards/premis/v2/premis-2-0.pdf

3.1. Standardizing the SIP

The ingest process handles two different kind of submission packages. The PANDAS package is significantly different from the package provided by Heritrix and the WCT. Content data is stored in a directory structure instead of using container files. The metadata provided by PANDAS is very limited compared to the metadata the WCT provides.

For reasons of efficiency the British Library decided to standardize the submission information package for the purpose of ingest into the Digital Library System (DLS). Standardizing the submission information package is also beneficial in the long term as it can assumed that different tools will be used for harvesting data and create Submission Information Packages. Modified policies and new use cases may require additional. As a consequence the container formats as well as the amount and granularity of metadata may change in the future.

Content and metadata are normalized when the sSIP is generated from the SIP. The sSIP supports a single container format. All content data must be embedded in one or more WARC[2] containers. The British Library decided to use WARC as the standard container for web content since it became a NISO standard in 2009.
ARC containers which are provided by Heritrix must be migrated to WARC containers. The web content harvested by PANDAS need to be migrated into WARC containers as well. In this case, the HTML files need additional transformations as PANDAS modified all the hyperlinks in the harvested HTML files. Instead of keeping the original URL, the links are using relative URLs pointing to the appropriate files the local file system. As the local files are embedded in the WARC container, the URLs need be replaced by absolute URLs.

Normalizing data and metadata from two very different sources is a challenge when both sources provide a very different quality of data. It becomes even more difficult, when future sources, preservation actions and requirements need to be considered. New tools may provide additional information or transform content in different, yet unknown ways. For this reason data model must be easily extendable and flexible to accommodate additional metadata.

3.2. sSIP/AIP content model

The standardized Submission Information Package (sSIP) uses a similar structure than the AIP. Metadata as well as content are stored in the same way using the same standards for metadata and container formats.
Both information packages share the same underlying content model. The content model defines so called abstract entities. Abstract entities represent the objects containing which need to be preserved. Unlike the content model of the web crawlers, the data model for the sSIP and

[2] Web Archive File Format: http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml

AIP disaggregates the preservation object from its digital manifestation.
Over time a number of different manifestations may occur. Content files will be migrated to new file formats. In abstract entity will remain and described by the same metadata record. Abstract entities are usually created by intellectual work. Therefore they are also called intellectual entities. Each of them may have a descriptive metadata record. This record describes the intellectual entity itself – e.g. the webpage and not its HTML manifestation.

The British Library's content for web archiving uses the following abstract entities:

- Website: a website is a collection of webpages which are interconnected and accessible under the same domain name. Usually the same organization or person is responsible for those pages. From the curatorial perspective a webpage must be regarded as preservation worthy in order to become part of a website. Not all webpages available under the same domain name become part of a website. A website must have basic descriptive metadata.
- Webpage: a webpage is a resource which is intended to be accessible and displayable as a distinct object. This resource is referenced by one or more hyperlinks which are forming the connections between webpages of a single website. Each webpage should have a descriptive metadata record. However in practice it proofed difficult to capture this metadata this metadata.
- Associated objects: An associated object is part of a webpage. Its rendition is embedded into the rendition of the whole webpage. The webpage provides the context for the associated object. An image being part of a webpage would be an associated object.

Every Webpage has a digital manifestation. It consists of at least one file or bytestream which can be interpreted and rendered to show the actual content of a Webpage. A digital manifestation may comprise several files. HTML based Webpages comprise of an html page, all referenced image files and Cascading Style Sheets containing important rendering information. An information system, such as a web browser, needs all those components to render a Webpage properly.

After a period of time this digital manifestation of a Webpage might become unrenderable. The manifestation or certain elements within the manifestation (e.g. the images) might not be rendered by common web browsers. The web as an open environment had to cope with incomplete support of standards as well as with different interpretation of standards from very early on. Complex Websites are often created for a certain group of browsers and browser versions. Once these browsers are not available anymore or unsupported by future operating systems, a migration might become a sensible alternative to provide further access to the content.

Whether a migration will result in a slightly different HTML file or a new file format (like PDF), it will create a new digital manifestation of a Webpage. Storing information about this migration process is essential for long term preservation. Preservation metadata attached to each file or bytestream must contain information about its origin. Providing an audit trail for a Webpage will ensure the authenticity of the data being stored.

A single manifestation consists of all content files which are required for a Webpage. All files are stored in WARC containers. For the initial crawl of a website a single WARC container will contain all files. The WARC container retains the curatorial coherence of the website.

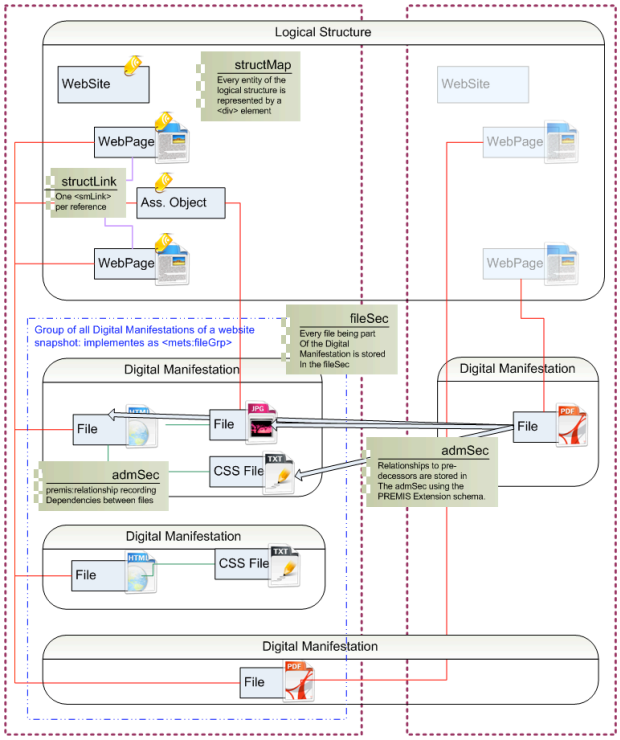

Figure 1. Content Model of the sSIP/AIP

The files of subsequent crawls or migrated Digital Manifestations may be stored across more than one WARC container. Every crawl will pass a deduplication process. This process detects any newly harvested bytestreams which had been stored earlier. In this case the file is not stored a second time. A migration might not impact all files. Some files of a digital manifestation might be unchanged. As no duplicates of files are stored, these unchanged files are stored in a separate WARC container than the migration results. In both cases the AIP's content is stored in several WARC containers.

3.3. Implementation of the data model

The data model was implemented in two stage. In the first stage the complete model was serialized using the METS framework. Every single object from the data model was represented by an appropriate XML element. As much metadata as possible were extracted, standardized and embedded into the METS descriptor including information

from the logfiles, the WARC container and WCT database. The resulting METS descriptor for a single SIP/AIP is very large.

In the second phase the METS serialization was reviewed regarding storage size. The underlying data model wasn't modified, it's serialization was. The main aim was to reduce the size of the overall SIP. This lead to a few basic principles:

- Only those objects are defined in the METS about something needs to be said: either because appropriate metadata records are available or these objects need to referenced, accessed etc.
- Information is just stored once where possible; metadata is not stored in the METS file if it as already stored in additional files (WARC container, logfiles) which are part of the SIP/AIP.

The size of the METS container could be reduced significantly.

```
<mets:structMap TYPE="logical">
<!-- the  website containing  webpages -->
   <mets:div TYPE="WEBSITE">

      <!-- the first webpage -->
      <mets:div TYPE="WEBPAGE"/>

      <! definitios of image -->
      <mets:div TYPE="ASSOCIATEDOBJECT" />

      <!-- the second webpage -->
      <mets:div TYPE="WEBPAGE" />

  </mets:div>
</mets:structMap>
```

Figure 2. <structMap> in METS representing the logical structure of a harvested website

3.3.1. Structural metadata

Abstract entities are represented by a element in the structMap section. The structMap section is the central section of each METS instance. Nested elements represent the logical structure of the website. According to the content model the hierarchy consists of three levels. For practical reasons the data model implements a simplified version of the content model. The data model's implementation restricts the hierarchical level of element to two. The uppermost elements represents the Website. All the other abstract entities are represented by elements which are direct children of the website's element. They are just available in cases there is something to say about the individual webpage or associated object (e.g. a metadata record is available). Otherwise the website's is the only in the structMap.

The content model defines relationships between webpages. They represent the navigational structure of a website and can be regarded as a site map. This information is also not recorded in the METS file for practical reasons. It would require parsing every html file to extract the links. Appropriate use cases justifying the additional effort are not regarded as relevant.

3.3.2. Descriptive metadata

Every element may have one or more descriptive metadata records. Each metadata record is stored in its own <dmdSec> element. The British Library's web archiving profile supports Dublin Core and MODS records. All the descriptive metadata related to the website such as the title and subject are mapped to MODS[1] using the latest 3.4 schema. Additional MODS elements such as typeOfResource, digitalOrigin and genre are set to fixed values. The Dublin Core elements being captured using the Web Curator Tool are recorded in a separate metadata section using the Dublin Core simple extension schema. Only those elements with very distinctive semantics are mapped to(e.g. title, creator) and recorded in the MODS record. Others with broad semantics such as dc:source are not mapable. These elements are only stored in the Dublin Core metadata section

3.3.3. Rights metadata

The web curator tool allows capturing basic rights metadata regarding the public access of data. This data is stored in the underlying relational database and used by the Access Tool when providing or restricting access to the content. As a consequence the rights metadata is proprietary. An extension schema had been derived from the WCT's database model. The rights metadata is recorded in its own administrative metadata section (<amdSec>) within the METS descriptor. Rights metadata are only available for the website. The administrative metadata section is attached to the element representing the website. Webpages or associated objects do not have their own metadata section containing access rights.

3.3.4. File defenitions

All container files are defined in the file section. Individual bytestreams within the container files are not defined unless there is a specific reason for it:

- A bytestream has a preservation metadata record which is needed in case of format migrations, recording certain provenance information etc.
- Deduplication: The content of a container belongs to more than one SIP/AIP; only relevant for the domain crawl, not for the selective crawl.

To distinguish the different purpose of bytestreams appropriate file groups are used. The web archiving profile supports the following groups:

- DigitalManifestation file group: The digital manifestation of all webpages and associated objects as well as their helper files (CSS, javascript etc.) are grouped into a single file group. It contains all files which are needed to render the whole website.
- Logfile file group: The Logfile group contains all logfiles. Created by the crawler (Heritrix or

[1] http://www.loc.gov/standards/mods

PANDAS). Logfiles provide useful provenance information. The crawl logfile contains information about every URL which had been requested. Error logfiles track any error which occurred during the harvesting process and may indicate the reason why a bytestream is not available in the AIP/SIP.

- Viral Files: all infected files are defined in this group; they are not regarded as part of the digital manifestation.

```
<mets:mets>
   <mets:fileSec>

      <!--
    The Digital Manifestation file group
     with the definition of two files -->
      <mets:fileGrp
              USE="DigitalManifestation">

      </mets:fileGrp>
      <!-- define a separate group for
            logfiles -->

      <mets:fileGrp USE="Logfile">
      </mets:fileGrp>

      < mets:fileGrp USE="ViralFiles">
      </mets:fileGrp>

   </mets:fileSec>
</mets:mets>
```

Figure 3. <fileSec> in METS defining different file groups.

As mentioned above all the data is stored in WARC containers. A WARC container consists of a so called WARC records - one for every successful http-request. Beside the actual content data it stores information on the http-protocol level such as response codes, file size and format type. Every WARC record is compressed within the WARC container.

In case a single bytestream needs to be defined in the METS, the complex structure of a WARC container must be represented. The container, the individual WARC record and the bytestream within the record are all represented by nested <file> elements.

The WARC record's <file> element contains specific information about the location of the WARC record within the container. The appropriate byte offsets are stored in the <file> element's BEGIN and END attributes. These two attributes were just introduced into the METS schema since version 1.9.

The same mechanism is used for recording the start and end of the actual content within the WARC container is stored in the content file's <file> element using the BEGIN and END attributes as well. It is important to note that before the actual content can be retrieved from the WARC record it needs to be uncompressed first. The <transformFile> element indicates which algorithm must be used for uncompressing the WARC record.

The value for the BEGIN and END attributes are also stored in the proprietary CDX files created by the Heritrix crawler. These files are index files for a WARC container and are used for randomly access content bytestreams from the WARC file. They are not part of the Archival Information Package as they are regarded as an access file. Its information can easily be reconstructed from the WARC container itself.

```
<!-- the WARC container itself -->
<file ID="container01">
  <transformFile
  TRANSFORMTYPE="decompression"
  TRANSFORMALGORITHM="WARC"
  TRANSFORMORDER="1"/>
   <!-- the WARC record within the WARC
        container -->
  file ID="gzip01" BETYPE="BYTE"
       BEGIN="20" END="22674">
    <transformFile
    TRANSFORMTYPE="decompression"
    TRANSFORMALGORITHM="GZIP"
    TRANSFORMORDER="1"/>

       <!-- the content bytestream within
          the WARC record -->

      <file ID="contentfile01"
      BETYPE="BYTE" BEGIN="623"
      END="35143" CHECKSUM="xxxxxx"
      CHECKSUMTYPE="SHA-512" SIZE="35123"
      MIMETYPE="text/html"/>
```

Figure 4. Example showing METS structure recording the internal structure of a WARC container file.

The first version of the METS profile defined every bytestream as a <file> element. The idea was to support (future) end-to-end business processes embedding all necessary information in the AIP. But as the current dissemination tool (wayback machine) doesn't support METS and the byte offset information can easily be extracted from the WARC file itself, the review regards the index information as redundant and consequently abandoned it from the SIP/AIP.

But having defined the mechanism for storing this information in METS, the AIP could record and provide all necessary information which is required for the dissemination of content. The data-requirements of the Access Tool had been considered when defining the AIP's data structure.

In case a bytestream is represented by a <file> element it must have an administrative metadata record attached. It contains basic preservation metadata as well as digital provenance information.

3.4. Preservation Metadata

The British Library's web archiving profile uses the PREMIS metadata schema as an extension schema to METS. PREMIS records are stored within each file's administrative metadata section (<admSec>). Content files, helper files and container files must have an administrative metadata section. Though WARC records are represented

by a <file> element they don't have a metadata record of its own.

3.4.1. Technical metadata

The preservation metadata record stores basic technical information about each file:

- Checksum: the SHA-512 checksum is calculated and recorded in the <premis:messageDigest> element as well as in the CHECKSUM attribute of the METS' <file> element.
- Size: The <premis:size> element records the size of the content bytestream in bytes. It contains the same information as the SIZE attribute of the METS' <file> element.
- Original URL: the URL which had been used in the http request. This URL is hostname based and may therefore not specify the actual server which submitted the bytestream to the crawler. In load balancing and virtual server environments the http-request may be redirected internally. The hostname based URL is recorded in the <premis:originalName> element and is retrieved from the crawler's logfile.
- File Format: The file format is retrieved from the HTTP-header in the http-response as it is recorded in the crawl log. The file format information is extracted from the crawl log and captured in <premis:format>. For the AIP this information is enriched with an appropriate reference to the PRONOM file format database using the DROID tool.

The METS descriptor stores preservation metadata for the container file as well. As the WARC file is assembled during the crawling process is does not have an original filename. The format information is set to "application/warc" as this is the official MIME type of the file format.

3.4.2. Provenance metadata

Provenance metadata is recording the history of a digital object. PREMIS provides an event framework for storing events within the bytestream's preservation metadata record. During a bytestream's lifecycle various events will have an impact on the object. Most events are occurring as part of well defined business processes. The METS profile defines all the events which may occur during the end-to-end process and have an impact on the web content during its life cycle. As business processes may change in the future, the event model may be extended with additional events.

Some events are only extracting or verifying information and don't have any impact on the bytestream itself. Other events have an impact on the bytestream as they are creating or modifying the content itself. To provide a comprehensive audit trail and ensure the information's authenticity it is important to record all events. Each event has a timestamp, an outcome and an associated agent. PREMIS defines an agent as a separate entity. Agents may be persons or software systems which were responsible for an event.

```
<premis:event>

    <premis:eventIdentifier>

      <premis:eventIdentifierType>local
       </premis:eventIdentifierType>

     <premis:eventIdentifierValue>event01
       </premis:eventIdentifierValue>

    </premis:eventIdentifier>

    <premis:eventType>migration
     </premis:eventType>

    <premis:eventDateTime>2006-07-16T19:20:30
    </premis:eventDateTime>

    <premis:linkingAgentIdentifier>

     <premis:linkingAgentIdentifierType>local
     </premis:linkingAgentIdentifierType>

    <premis:linkingAgentIdentifierValue>
      agent001
     </premis:linkingAgentIdentifierValue>

    </premis:linkingAgentIdentifier>

</premis:event>
```

Figure 5. Representation of an event in PREMIS

The web archiving profile defines two different events: virus check- and migration event. The harvest process is not recorded in the PREMIS record as metadata should not be stored redundantly. The logfiles whicha re part of the SIP and AIP are already containing appropriate provenance information such as the URL, IP-address, datetime stamps and the http-return code.

Virus-Check Event: Each file which is ingested into the archival store is checked for viruses. The virus check event is recorded on level of the WARC file. Only in the exception that a virus had been detected, the appropriate bytestream is defined in the file section and an appropriate PREMIS record with the event information is recorded on bytestream level.
In case a virus is detected the ingest system tries to clean the effected bytestream. This may or may not be successful. Depending on the success, the output of the event is recorded in the <premis:eventOutcome> element:

Value for eventOutcome	*Virus check outcome*
no virus detected	No virus detected
viral, cleaned	virus detected, bytestream had been cleaned successfully
Viral, failed but forced	virus detected, but cannot be cleaned

Table 2. Event outcome values for the virus check event

The <premis:agent> element records the anti-virus software and its virus database version which had been used during this event
In case the viral file could be cleaned, the original, viral file is stored in a separate file group marked as "viralfiles". It is not part of the website's digital manifestation. The viral file is only kept for administrative purposes and will not be accessible by end users via the Access Tool. Instead the cleaned file will be part of the website's digital manifestation file group.
To keep track of the bytestream's history the relationship between the cleaned and infected bytestream is kept in its preservation metadata record. The <premis:relationship> element records a pointer to the old viral bytestream. The relationship type is set to "derivation" and its subtype to "cleanedFile".

Transformation Event: HTML bytestreams which had been harvested using PANDAS need to be transformed. All URLs need to be updated. This transformation process takes places prior to ingest into the Archival Store. The appropriate transformation event is recorded as part of the standardized Submission Information Package as well as in the Archival Information Package.
Rewriting the URLs results in a set of new HTML files. Though both sets of HTML bytestreams are defined in the METS descriptor, only the new HTML bytestreams are part of the Digital Manifestation. The original bytestreams are part of a separate file group. Image bytestreams, style sheets etc. are just part of the same Digital Manifestation as the new, transformed bytestreams.
The transformation and the relationship between the old and new bytestream are recorded in the ew btyestream's preservation metadata record .

Element name	*value*
premis:relationshipType	Derivation
premis:relationshipSubType	Transformation

Table 3. Relationship between PANDAS html files and transformed html files.

A transformation is regarded as successful, whenever the new bytestream exists. The <premis:eventOutcome> Element can only contain the value "success" for the transformation event.

Migration Event: The Archival Store supports WARC as the only container format for web content. All ARC containers are migrated into WARC containers prior to ingest. This migration process is described in the WARC-container's preservation metadata record.
The outcome of this event always "success"; otherwise the WARC container would not exist. A relationship between the WARC and the ARC file is described using the <premis:relationship> element pointing from the WARC file's to the ARC file's preservation metadata record.

Element name	*value*
premis:relationshipType	Derivation
premis:relationshipSubType	Migration

Table 4. Relationship between WARC and ARC container files

When container files are migrated, the actual content bytestreams stay untouched. Consequently event information for individual bytestreams is not recorded.

4. CONCLUSION

The three different information packages which are defined by the OAIS are used for three very different purposes. Though the British Library's METS profile for Web Archiving does not define a Dissemination Information Package, it supports the end-to-end business process of harvesting, ingesting and preserving web pages and enabling long term accessibility. The METS profile considers access as well as preservation requirements.
The practical implementation does not make use of the whole complex data structure: METS files become fairly large and a lack of support of METS by dissemination tools makes it inconvenient and expensive to store all the metadata redundantly. Instead the profile ensures that all the metadata is part of the SIP/AIP – either as part of the METS descriptor itself, as part of a proprietary file (logfiles) or embedded in and restorable from the actual content file (index of WARC files).

Using standardized metadata frameworks and schemas such as METS and PREMIS are as important as an extendable and flexible content model. Defining abstract entities and their manifestation as separate objects allows future implementations of tools to support a complex end-to-end process without relying on proprietary data structures.

It ensures easy integration of the web archiving content with other content streams and library systems. As the operational and archival metadata is now being managed in the library's preservation repository content can actively be preserved. Preservation actions can be carried out, new digital manifestations of web content can be created.