The International Journal of Digital Curation
Volume 7, Issue 1 | 2012

A Short Story about XML Schemas, Digital Preservation and Format Libraries

Steve Knight,

Programme Director, Preservation Research & Consultancy,

National Library of New Zealand, Te Puna Mātauranga o Aotearoa

Abstract

One morning we came in to work to find that one of our servers had made 1.5 million attempts to contact an external server in the preceding hour. It turned out that the calls were being generated by the Library's digital preservation system (Rosetta) while attempting to validate XML Schema Definition (XSD) declarations included in the XML files of the Library's online newspaper application Papers Past, which we were in the process of loading into Rosetta. This paper describes our response to this situation and outlines some of the issues that needed to be canvassed before we were able to arrive at a suitable solution, including the digital preservation status of these XSDs; their impact on validation tools, such as JHOVE; and where these objects should reside if they are considered material to the digital preservation process.

International Journal of Digital Curation (2012), *7*(1), 72–80. http://dx.doi.org/10.2218/ijdc.v7i1.215

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

The Initial Cause For Concern

Server has made 1.5 million attempts to contact 217.111.76.181 in the last hour, stopped by our local firewall. What does this IP map to? Why is it doing this?

What Was Happening?

We were testing the legacy Papers Past[1] migration process, which we were undertaking in order to bring 61 titles (300,000 issues, 4.2 million files, 3.6TB) into the Library's digital preservation programme. This included ALTO XMLs[2], many of which included the XML declaration:

```
xsi:noNamespaceSchemaLocation=http://schema.ccs-gmbh.com/metae/alto-1-2.xsd
```

The address 217.111.76.181 translated to the host: schema.ccs-gmbh.com. As part of the format validation and metadata extraction, an application (it turned out to be JHOVE) was attempting to make the internet connections from the deposit server in order to download the XSDs and validate the ALTOs against the corresponding XSD.

An analysis of the XML (ALTO and METS) in the entire Papers Past repository was undertaken, which took about three days and resulted in the following list of XSDs referenced from within the Papers Past repository:

- http://schema.ccs-gmbh.com/docworks/alto-1-2.xsd
- http://schema.ccs-gmbh.com/docworks/mets-metae.xsd
- http://schema.ccs-gmbh.com/metae/alto-1-2.xsd
- http://schema.ccs-gmbh.com/metae/mets-metae.xsd
- http://www.loc.gov/standards/mets/mets.xsd

Note that, on some occasions, an XSD will include within it pointers to another XSD.[3],[4]

What To Do?

Our first instinct, and easiest choice, was to open up the appropriate port to allow the connection to the internet and therefore the validation of the ALTOs against the appropriate XSDs at their home location.

[1] Papers Past: http://paperspast.natlib.govt.nz

[2] Alto is most often used as an extension to the METS standard for describing the structure of objects. The ALTO extension provides more granular detail related to the content and layout of each page of an object, e.g. books, journals, newspapers. ALTO: http://www.loc.gov/standards/alto/

[3] XML Schema: http://www.w3.org/XML/Schema.

[4] XML Schema Tutorial: http://www.w3schools.com/schema/default.asp.

However, there are a couple of underlying philosophical issues here which needed addressing.

Firstly, how much intervention is allowable or preferred to ready a document for ingestion into the permanent repository? None, or as much as you like? Like most organisations, we have regularly spun around this pole. At one end of the continuum we cannot afford to be too precious in pre-ingest activities, which are designed to make an object preservation-ready. This could include editing out unwanted material, which adds no preservation value or comprises no internal value to the Intellectual Entity, but which impedes ingest. At the other end of the continuum, such editing could be seen as tampering with source files and therefore the authenticity and integrity of the objects involved.

Editing the XML itself to change the reference location in the XSD also raises its own issues:

- There are approximately 2.4 million XML files, which means there are implications for storage and infrastructure requirements for processing such a large corpus of material;
- The potential to introduce other issues (e.g. unnoticed errors) into the XML;
- The question of how to validate the accuracy of any changes made;
- The question of when in the workflow process to make changes to XML: whether at the source location or at the time of metadata extraction, each of which has its own set of issues;
- The potential impact of changes on the Papers Past application, website and delivery, each of which would require its own analysis and testing following any changes to the XML data.

The second key question raised by this issue is related to the role of an XSD declaration. Does an XSD have any preservation value? If so, what is it and how does our current process help us to elucidate that value and resolve any associated issues that may arise?

There has always been an understanding within the National Digital Heritage Archive (NDHA) team, which is responsible for the National Library of New Zealand’s digital preservation programme, that ancillary documentation regarding formats, e.g. format specifications, should be kept as part of the programme, and that the right place to hold these objects was in the Format Library. Why would XSDs be any different?

Wikipedia defines an XSD as:

“...a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself...”

And:

"...the process of checking to see if an XML document conforms to a schema is called validation, which is separate from XML's core concept of syntactic well-formedness." (Wikipedia, 2012)

Consequently, in the context of a long-term preservation programme, it made sense to us that the XSD (the validating document) is kept with the originating XML document (the document to be validated). So that is what we decided.

The XML Catalogue: Where to Locate and How To Access The XSD?

Having determined that the XSD is a preservation attribute of the object being ingested into the permanent repository, the next question to be canvassed was where should that object reside. Clearly, the object has been designed, via the schemaLocation hint, to access the appropriate XSD at the host named in the declaration. But, is this satisfactory in the context of a long-term preservation programme? What certainty do we have that the host will remain available and accessible? What certainty do we have that the XSD itself will remain available and accessible at that host?

Having considered these issues, we decided that a local XML catalogue for this type of material was the best approach. The Library had prior experience of this as part of its digitisation programme. An XML catalog provides a standardised way of mapping "external identifiers and URI references to (other) URI references … the principal task of a catalog processor is to find entries in the catalog that *match* the input provided and return the associated URI reference as the output. The first such match is always used, and there is no requirement for the catalog processor to search for additional matches." (OASIS, 2001) The key benefit of such an approach is the ability to keep point-in-time versions of XSDs as part of the Rosetta implementation and subject them to regular, managed backup under our control. It is worth noting here that while the specific issue being addressed by the Library related to XML Schema Definitions, the XML catalog approach is also applicable to any Document Type Definition (DTD).

Having made this decision, the next question was where to store the catalogue. The Library's digital preservation system, Rosetta, does not currently have the functionality within its Format Library to hold this type of documentary material, which we have determined as imperative to the preserve-ability of the objects in the permanent repository. The same applies to format specifications mentioned above.

The short term solution is to provide an internal location for this material to reside. It is our expectation that as the Rosetta Format Library evolves it will do so in a manner that will cater for this type of documentary evidence.

The XML Catalogue Approach requires less change overall, comprising:

- One time enablement of the catalogue within JHOVE,
- One time creation of the catalogue XML file to provide the translation mechanism,
- Providing a local site for the catalogue and the relevant XSDs within the Rosetta environment.

There is an overhead to this approach. When any new XSD is introduced (e.g. ALTO 1.3), the Library will need a technology watch function to recognise and respond to this through the updating of the catalogue file within Rosetta. However, this overhead does have the added benefit of retaining control within the digital preservation programme. As the XML catalogue approach does not require opening a new port, as any reference to a new, uncatalogued XSD will be flagged up at the firewall, all else failing.

JHOVE:
Will it Support the XML Catalogue Approach?

As the entry point into this issue was the JHOVE validation components, we decided to test our thinking against the team at California Digital Library (CDL), the project lead for the JHOVE2 development currently underway.

That discussion:

- Confirmed JHOVE1 does not support XML catalogues,
- Validated NDHA's customised code development within JHOVE1 to enable XML catalogues,
- Confirmed JHOVE2 does support XML catalogues (JHOVE2, 2010).

The discussion with CDL also affirmed both the value of XSDs as preservation attributes of the objects that reference them, and also that an XML catalogue was a valid approach. The following from Stephen Abrams at CDL:

> "I think that your instinct to capture and locally manage all DTDs and schemas referenced from XML content object is correct... Using a catalog seems to be preferable, since it adds a layer of external indirection that will tend to decouple the referencing mechanism from details of repository design and implementation." (Stephen Abrams, personal communication, 7 August 2011).

XSDs and Enhancing the Digital Preservation Process

There is an emerging concern within the NDHA that the level of record-keeping for digital preservation is not as sophisticated or comprehensive as it should be. For example, as we move towards a local XML catalogue solution to externally and internally referenced XSDs, the digital preservation system needs to know:

- Is there a provenance implication, and if so what is its nature and extent?
- At which level of the hierarchy should a provenance note be attached, e.g. Intellectual Entity or file level?
- What are the implications for the data model of either of these choices?
- Having used an internally customised version of JHOVE (providing the XML catalogue functionality) how will we understand which material has been validated by JHOVE1, NLNZ-customised JHOVE and JHOVE2?
- If we switch back to JHOVE1 from our NLNZ-customised JHOVE, how will we measure and manage any subsequent impacts on material already validated by the customised JHOVE?
- What are the data model implications of having material in the permanent repository validated by more than one version of JHOVE?
- Should there be a facility to run more than one version of a tool (e.g. JHOVE1, NLNZ-customised JHOVE and JHOVE2) or different types of a particular tool, such as ClamAV and ESET Nod 32 for virus checking? (Note that Rosetta will be able to cater for multiple virus checkers in an upcoming release).

As it turns out, the customised JHOVE (i.e. supporting XML catalogue) cannot put any additional elements into the file's data model. To include this would require changes to the JHOVE plugin in the Rosetta code. Nor can we add a provenance note into the file data model within Rosetta. Plugins are not designed to insert provenance notes. This can be worked around at metadata extraction time, but this is less than ideal.

The Solution

On ingest of XML files into Rosetta, we would like JHOVE to validate them against their XSD declarations. Rather than attempting to locate the XSD outside the NDHA, we want the XSD to be located locally within the Rosetta environment, if not yet supported through the Formal Library. So, how did we configure JHOVE to locate this XSD under an alternate location?

In both staging servers of the test environment we replaced the jhove-module.jar under /exlibris/dps/d4_1/system.dir/thirdparty/jboss/server/default/deploy/repository.ear/lib/ with a JAR version that is XML-catalog-enabled.

The following catalogue files and the downloaded schemas were copied to appropriate directories:

- CatalogManager.properties: /exlibris/dps/d4_1/system.dir/conf/
- xml_catalog.xml: /net/unua/natlib_data/elscratch/XmlCatalogues/
- A bunch of downloaded schema files: /net/unua/natlib_data/elscratch/XmlCatalogues/schema_files.

We then restarted both staging servers.

The Results

- Customised JHOVE was able to locate the initial XSD reference to the local copy of the referenced schema file (e.g. alto-1-2.xsd, mets-metae.xsd).
- Customised JHOVE failed to locate the other XSD files that the initial XSD was referencing internally.
- The XML file inside the test SIP failed as "not valid", probably as JHOVE was unable to locate XSD files referenced in the original XSD.
- Fortunately, it turned out that references to these other XSDs were inputted into the catalogue incorrectly. This was subsequently fixed.
- The XML file inside the test SIP was accurately tested as "valid".

Where to From Here?

As noted above, the primary decision making here was around the preservation value of the XSDs and their role as providers of assurance regarding the integrity of the XML document from which they were referenced.

However, the real issue relates to the ongoing development of the Format Library and the nature of the services it supports. From a digital preservation perspective, it is preferable to maintain relationships with elements such as .XSD via a local .XSD catalogue rather than externally.

It also makes sense that that local instantiation should occur within the Format Library, which is where you'd expect documentation, specifications, schemas, DTDs and other supporting file formats to reside.

It seems unlikely that this will be an NDHA problem only and this has since been confirmed via the conversation with CDL.

As such, it would appear sensible for this to be embedded in the ongoing development of Rosetta and particularly within the broader specification of the Format Library as the core support for the preservation processes.

This would require Ex Libris assistance in two ways:

- Firstly, in support of JHOVE (i.e. the new JHOVE2 catalogue functionality. While JHOVE2 can be installed to Rosetta, it is not yet out-of-the-box) and in particular the ability to run and manage more than one version of JHOVE in the system;
- Extension of the Format Library to support a wider view of the technical and documentary landscape within which file formats exist and co-exist;
- Extension of the data model to include more granular information regarding multiple JHOVE usages.

As noted above, this level of functionality will be provided in an upcoming release of Rosetta in the context of virus checking. As a principle it would be good to see this type of flexibility in regard to all third party offerings, like JHOVE, along with a commitment to being able to run multiple versions of JHOVE with the associated changes to the data model to support better record keeping regarding the use and management of third party tools, such as JHOVE (preferably at both file and intellectual entity level, i.e., the individual image or article or issue of a newspaper).

Conclusions

While the Library's resolution of the particular situation relating to XSDs does not entirely resolve for us the primary philosophical issues regarding how much intervention is allowable or preferred to ready a document for ingesting into the permanent repository, it has helped us to develop a framework within which other such issues can be discussed and resolved, e.g. changing original filenames, changing or providing file extensions when they are wrong or are missing from the original object, and so on.

It has also forced us to face the issue of record keeping in digital preservation, the preservation value of ancillary documentation, such as format specifications and XSDs, and possibly most importantly, the central role to be played by the Format Library in tying together all the varied aspects that comprise a fully realised knowledge base for the ongoing execution of preservation planning and risk management.

Acknowledgements

With thanks to my colleagues Raghu Pushpakath and John Barris, who kept us grinding away at a solution to a very interesting problem.

References

JHOVE2. (2010). JHOVE2: Next-Generation Architecture for Format-Aware Characterization: XML Module, Version 0.4. Retrieved from https://bytebucket.org/jhove2/main/wiki/documents/JHOVE2-XML-module-specs-v04.pdf.

OASIS. (2001). XML Catalogues: Committee Specification 06 Aug 2001. Retrieved from http://www.oasis-open.org/committees/entity/spec-2001-08-06.html

Wikipedia. (2012). *XML Schema.* Wikipedia, The Free Encyclopedia. Retrieved from http://en.wikipedia.org/w/index.php?title=XML_schema&oldid=473478988.