Archiving of XML, Image Data and the Visual Representation in Form of a PDF

Jeremias Märki, Matthias Günter, 2006-02-21, Version 0.7

Abstract

This paper discusses different ways to package a document in its final form (ex. PDF for long term archiving) together with the original data (ex. XML plus images) that was used to produce the final form.

Introduction

For many documents the form it was printed / published in is important information that needs to be archived[1] as well. In the patent community and in other areas citation is often based on page number, which in electronic documents is inadequate. On one side there is the visual representation that should be archived and on the other side there is the storage of the semantic information in the form of an XML (with inclusion of images and other embedded information.). The embedded information must be stored carefully, so it does not create obstacles in long term storage. Until now patent documents are stored as TIFF-images since 1970 with obvious problems in search and retrieval.

For archiving PDF documents, PDF/A was defined by IS-O (in ISO 19005-1).

Signing the document electronically for integrity is a useful instrument as well, but should be done in a way to not pose problems with long term archiving. Digital signatures so far have not proved to be retrievable or useful in long term storage.

Current problems

The current problem for a user was that he or she either had structured data or data ready for presentation. Combined approaches only exist for native applications and even then the look may change with each update of the software. Only when keeping the computer, the application and the data you can be sure that a document can be reconstructed. To have a reliable graphical representation could be done with images (e.g.) and lately with PDF/A. However, the data become unstructured at best with such an approach.

A reliable and reproducible version of both the representation and the underlying data is a need in long-term archiving, but also helps with an easy way for communication to humans and machines in one file. Keeping everything in one file helps house-keeping in the ever-increasing data sets.

Approaches with some kind of preconditions by standardisation bodies

The following approaches are for the time being impossible, because the relevant standards don't allow for them. However, they are both easily be made available and certainly work out. The XMP/RDF approach has a special elegance.

[1] More than 20 years.

Embedding the XML as an embedded file stream in PDF

This approach is invalid as it is forbidden by the PDF/A-1 specification. However, if an exception would be made for archiving XML-file streams then it is one of the solutions that is covered by this paper. It is done easily enough.

Using XMP/RDF Metadata in PDF

The XML is stored into the XMP[2] metadata of a PDF/A-1. Unfortunately this does not allow for PDF signatures (which need PDF 1.5). PDF/A-2[3] will support this (based on PDF 1.6). In the case of patents images (JPEG/TIFF) are included as well. These are forbidden in PDF/A-1. The XML schema or DTD can be changed to include the images in encoded form (uuencode, binhex, base64 or others)[4].

Images can also be extracted from the PDF image objects (by or without including meta information on the images) and be added to the XML again.

A simple RDF-Schema can be defined and the original XML is embedded as resource (http://www.w3.org/TR/REC-rdf-syntax/#xmlliterals). However, XMP only uses a subset of RDF which disallows the use of „rdf:parseType='Literal'“ which makes embedding the original XML data impossible. Embedding the original data in the form of an RDF structure is equally impossible since RDF doesn't support mixed content.

If a future version of XMP would allow it, it would be no problem to store general XML content in the XMP.

---

[2] http://partners.adobe.com/public/developer/en/xmp/sdk/xmpspecification.pdf

[3] Information on work: http://www.aiim.org/standards.asp?ID=25013

[4] For possible examples see: http://uk.builder.com/manage/work/0,39026594,20268607,00.htm

Custom Schemas in XMP/RDF Metadata in PDF

It would be possible to create new XMP schemas with the extension mechanism and use it for the storage of information. As patents of different national offices have different formats and Schemas/DTD, this solution is rather inelegant and could be used for bibliographic information at most without compromising the approach.

Embedding as value in PDF

Within the XMP/RDF framework the whole XML representation including additional elements like images could be coded (uuencode, binhex or other) after being compressed or not and stored as a simple element/variable within the XMP/RDF meta data instead of forming a full stream. This approach works but is not elegant at all.

This could be done by a custom new XMP schema or by using an existing element.

In the end event, dc:source (from Dublin Core) might be used for it.

Possible approaches

Possible approaches are the ones that can be made easily within current restrictions by standards.

Method 1: ZIP/JAR

The XMP approach above would have added some amount technical complexity to extract the XML from the PDF and also the restrictions of PDF/A-1 apply. A ZIP-based approach allows to use a standard method. The only drawback is that the PDF must be extracted to view it.

In the ZIP-based approach each document is put into a ZIP archive. The root documents are assigned predefined names, e.g.

- main.xsd or main.dtd
- main.pdf
- main.xml

All other files and data can be added as well to the ZIP and referenced from the XML file.

Signing can be done with the JAR-mechanism[5] or a different method. Other compression/archiving methods can be used as well.

Method 2: OpenDocument-Like

The method bases on method 1, but uses an OpenDocument-like structure[6] of the compressed file.

As MIME type „application/archived-xml-pdf“ is proposed, if accepted by IANA. Instead “application/X-archived-xml-pdf” can be used until then. Because we are talking about long term storage here, possible embedded items in the XML are restricted as well. For patents these are images which must meet stringent requirements[7].

---

[5] http://java.sun.com/docs/books/tutorial/deployment/jar/signindex.html

[6] http://de.wikipedia.org/wiki/OpenDocument

[7] http://www.wipo.int/pct/en/texts/pdf/ai_anf.pdf

For the main documents we suggest again main.xsd, main.dtd, main.xml and main.pdf.

The signatures can be stored according to the JAR structure[8] or according to other methods, including those that OASIS will develop.

Method 3: General Multi-Document Packaging based on Method 2

Method 2 can be generalized to make the packaging of multiple representations of a single document possible, without directly restricting the use case to XML + PDF. For example, it may make sense for other applications to package a PDF or HTML file together with the original Word or OpenDocument document that the derived file was created from. This means we need a mechanism in the ZIP package to describe the individual files. This can be done in a manifest file similar to the one used by OpenDocument. The manifest file would list all the primary files (the PDF, the XML or the OpenDocument file). Under some primary files nested elements will mark the resources used by the primary document if any. Example: An XML primary file could reference the images it needs. Each file specifier will carry a role attribute which may have one of at least three possible value: “presentation”, “original” and “resource”. This list may be further specified by a follow-up specification of the package format proposed here. Every file placed in the package will need to be represented in the manifest file. Also, every manifest entry will allow for multiple (0..n) cryptographic signatures, so the integrity and origin of the whole package can be verified. To get a quick overview over the contents of the package we can provide for a possibility to embed XMP metadata within the manifest file.

In our case here, we can specify certain restrictions to the use of the general package format, so we can directly apply it to the storage of patent documents.

Method 4: Separated

PDF and XML are in separate files. The XML are packaged together with the auxiliary documents in a document according to method 1 or 2. The signature for the PDF is also stored in that document. A special tool is then needed to verify the signature of a PDF file. For the presentation to the user the PDF/A could be converted to PDF 1.6, the signature added and then verified by the user. However, the conversion process will make the PDF 1.6 compliant file a derived file. It's not the original anymore.

Standardisation

The goal is to create a generally accepted container method (by standardisation) with defined access methods. In this case all applications that want to work with the container (reading and/or writing and verifying) will use a simple API to access the data.

We realize that there are other efforts elsewhere (some listed below) to define similar package formats, sometimes as part of a bigger work. However, most are too complex for our situation or are still in development. Since we are under time constraint and try to reuse existing technologies (like XMP and ZIP) where possible a migration to another system should easily be possible at a later time.

---

[8] http://java.sun.com/j2se/1.3/docs/guide/jar/jar.html#SignedJar-Overview

Other Graphical Representations

Instead of PDF other graphical representations can be chosen. RDF, Dublin core, XMP and other can be used with several other graphical representations (e.g. TIFF, JPEG etc.). There are also other container formats that can be used. In each case the structured representation can be inserted by one of the previously stated methods.

Other Structured Representation

Instead of XML all other possible structured data formats can be used. ASCII, CSV, key-value-pairs etc.

Multiple representation

It is possible to use several graphical and/or structure representations to achieve the goals. It might e.g. be wise to include XML but also an ASCII representation of the content.

Further Reading

MetaData - Models Compared: http://www2-data.informatik.unibw-muenchen.de/LZA/Seminar_DocEng_HT2004/t21_Vortrag.pdf

OAIS - *Open Archival Information System: http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html*

Projekt „kopal“: http://kopal.langzeitarchivierung.de/