



---

**Integrating automatic metadata generation into deposit workflows  
(Web Service orchestration)**

**Richard Green**

**June 2009**

## **Integrating automatic metadata generation into deposit workflows (Web Service orchestration)**

### **Why is this important?**

One of the major purposes of most digital repositories is to allow users to search for and to discover content. Whilst for textual content these two processes might be enabled through full-text indexing, for other content quality metadata is key to the process; even in the case of textual materials there may be aspects of the digital content that should be represented but which do not lend themselves to indexing in a conventional manner. We might distinguish three types of metadata for the purpose of this report:

- descriptive metadata which deals with the digital content of a repository object (sometimes called 'resource discovery metadata')
- administrative metadata which deals with administrative and technical matters
- preservation metadata which helps inform the management and potential long-term preservation of a digital object (this and administrative metadata are not necessarily disjoint categories)

Metadata, then, is of central importance to a repository but at the same time adding metadata to digital materials is often seen as one of the less agreeable aspects of working with repository content. All too often the creation of metadata is seen by content creators as a tedious task bolted on to the end of the creative process. How much better, then, if the creation of metadata could become an integral part of that process requiring little additional work by the creator.

This report examines how metadata creation could be an integral step in the process of ingesting digital content to a repository and it considers how much of this metadata might relatively easily be generated automatically. Further, it considers briefly how the inclusion of some additional metadata might assist repository managers in the management and possible preservation of the digital content once ingested to the repository.

This report draws heavily on the author's experience managing two JISC-funded projects, RepoMMan (2005-2007) and REMAP (2007-2009) for Academic Services at the University of Hull. All of the ideas here have been implemented at least to proof-of-concept level; some have been taken well beyond that and are being piloted at the University with a number of staff. The work has also been taken up by the Hydra Project, an international collaboration between the University of Hull,

the University of Virginia, Stanford University and Fedora Commons (this last the not-for-profit organisation that provides the open access repository software used by the other three institutions).

The approach taken in Hull, and now by the Hydra Project, involves the use of Web Services. To quote Wikipedia:

*A Web Service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network". Web services are frequently just Internet application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.*

Web Services potentially offer users a number of advantages, amongst which interoperability between various software applications, quite probably from different providers, running on disparate platforms and in disparate locations. In addition, Web Services of appropriate granularity can potentially be re-used in contexts other than the one for which they were originally developed. It should be noted that Web Services can also be executed on the local computer and, as we shall see, a mixture of local and remote Web Services may be appropriate.

The Fedora repository software was designed from the beginning to be used and interacted with via its Web Service APIs and for many years has supported SOAP (Simple Object Access Protocol) interaction: the most recent versions of Fedora additionally support a full REST (Representational State Transfer) Web Service interface. In all but the simplest of scenarios, Web Services require some measure of 'orchestration' to call them in an appropriate sequence (a process that may involve conditions, loops etc) and the work at Hull used BPEL (the Business Process Execution Language, an open standard) to do this; a number of other orchestration tools exist.

Up to this point, this report has avoided use of the term "workflow" which appears in the title. The reason for this is that in the repositories community there is often a lack of consistency in the way the term is used. Some talk about workflow in broad terms: the sequence of steps undertaken by a human to complete a particular task. Others may use the term in the context of sequencing Web Services to achieve a task. For the remainder of this report any use of 'workflow' will relate to the broader human activities and the term 'orchestration' will be used to describe the sequencing of Web Services.

## Some Scenarios

### Scenario 1: Depositing a thesis into a repository

Ruth has completed her PhD thesis and the final version has been approved by her examiners. The regulations at her university require her to deposit a digital copy of the thesis in its institutional repository. The thesis is a single, straightforward Word file with no additional components. Ruth has used a "My Repository" facility offered by her university to function as a digital vault during the later stages of her work. The use of this private repository space has also given her easy control over document versioning as her thesis evolved. The tool that she uses to manage the contents of this repository space offers a button labelled 'publish' which will transfer a copy of her work into the institutional repository's mediated publishing queue. Using this publish process Ruth is presented

with a wizard that guides her through the process of adding appropriate metadata to her thesis; a level of automation is provided so that, rather than being presented with blank forms in which to insert the metadata, forms are pre-filled where possible. Ruth can choose to accept the metadata offered or to edit it.

The tool that Ruth uses to manage her private repository space is an Adobe Flex application surfaced in her web browser. The application interacts, using SOAP Web Service calls, with a BPEL server elsewhere in the university. This BPEL server manages all the Web Service interaction with the tool from simple calls to more complex, orchestrated sequences of Web Services required to carry out complex tasks. When Ruth first invokes the tool, a copy of her thesis is taken from her private repository space. It is sent to a Web Service that analyses the content of the file to extract descriptive metadata. This Service exposes aspects of the iVia metadata software described in 'Tools' below. The wizard then asks Ruth what kind of text document she is submitting and, when she indicates that it is a thesis, the output from the tool is assembled behind the scenes using the appropriate metadata format (in this case, UKETD\_DC: see 'Standards' below). A Web Service is also called that extracts technical metadata about the document, for instance the file (MIME) type and the file size; this is based on the JHOVE tool from Harvard, again described below. Ruth is then presented with a short sequence of screens which enable her to provide metadata to describe the thesis that she is submitting. Many areas of the screens are pre-filled using the metadata extracted from the file: these she can just accept, or she can correct or extend them. Some aspects of the metadata (for instance her supervisor's name) will not have been identified by the metadata tools and these she has to fill in. When the metadata is completed to her satisfaction she can save it as part of the thesis object in her private repository space, she can also invoke a final 'publish' sequence which uses an orchestration of Web Service calls to build a completely new digital object around her content. This new object conforms to a standard structure for theses within the institutional repository, and is placed in the institutional repository's mediated accession queue. Part of this process is a Web Service that will convert her Word file to a PDF file, the University's preferred repository format. Once checked by a member of the repository staff it is this object that provides the metadata and PDF file for public use; hidden components of the object contain a preservation copy of the original Word file and possibly preservation metadata such as will be discussed in scenario three.

### **Scenario 2: Depositing an image into a repository**

Mike works as a photographer for the marketing department of his university. He is frequently asked for general images of the campus that staff or students could include in documents that they produce. He responds by setting up a small collection of such images in the university's institutional repository.

Today Mike wishes to add a new image to his collection; he is able to use the same Flex tool that Ruth used in 'scenario one'. He uploads the image that he wishes to place in the institutional repository into his private repository space, highlights it and clicks the publish button. The tool asks what sort of image Mike is depositing to which he responds 'digital photograph' by selecting the term from a drop-down menu. This choice determines the detail of the process that follows. Like Ruth, Mike is then led through a short series of screens that gather metadata about the image.

Where possible the forms are again pre-populated although Mike will be required to provide rather more description than was Ruth as there is no tool for generating descriptive metadata for an image. Web Services are invoked behind the scenes to populate, in this case, a Dublin Core metadata record. In addition a thorough technical analysis of the image file is undertaken and the technical metadata recorded within the digital object created for the repository; the information includes the camera EXIF information for the photograph, often of interest to serious photographers comparing images. As part of the orchestration that constructs the digital object for the institutional repository, not only is Mike's original image stored but also a number of surrogate images at different resolutions. As in Ruth's case, the digital object is placed into the mediated accession queue for the institutional repository.

### **Scenario 3: Depositing minutes from University Committees into a repository**

Margaret has responsibility for lodging the minutes of several statutory committees in her university's institutional repository. These minutes are required to be kept 'for ever' and the university's repository managers must try to ensure that they are just as accessible in 50 or more years' time as they are now. In that period, digital technologies will clearly have moved on and it is likely that the PDF files she has stored would no longer easily be used. It would seem likely that they will need to have undergone a number of format migrations along the way to ensure that they remain accessible to potential users.

Margaret uses an enhanced version of the Flex tool described above to deposit each set of committee papers into the institutional repository. This enhanced version potentially embeds three additional forms of preservation metadata into the digital object created. The first is the output generated by a local implementation of the DROID tool developed by The National Archives (TNA) in the UK. The tool is exposed as a Web Service and, when called as part of the orchestrated deposit process, analyses in detail the file format around the minutes that Margaret is submitting. This file signature is then tested against TNA's PRONOM tool which may recommend suggested preservation activities for the format. The PRONOM return is stored for reference. A third element of metadata is stored in the object which triggers a periodic check of the DROID metadata against PRONOM; a file format that is deemed 'safe' at the moment may be considered 'at risk' next year. This periodic check is one of a number of management and/or preservation processes which may be initiated by automated checking of metadata stored in the digital object. Margaret's institutional repository has been set up in such a way that each digital object within it has metadata capable of triggering specific activities over time either by invoking automated processes or by alerting an appropriate human to the need for some action; in this second case there may be an automated default action if the human does not respond.

In Scenarios 1 and 2, the Web Services described were provided by the university itself. There is no technical reason why this should necessarily be so but local implementations seem preferable to transferring large volumes of data around the internet unless there is a compelling advantage to that. Clearly, though, it will be the nature of some of the Web Services that their local instantiations may need to be updated periodically for instance, the file signatures used locally for DROID testing can be remotely updated from TNA using a dedicated Web Service. The PRONOM software at TNA is regularly updated to take account of the changing status of file formats. If it were installed locally it

would need to be updated with some frequency. As the DROID output that it consumes is relatively very small, it makes sense to send this for analysis by the Web Service exposed at TNA and thereby to take advantage of the latest PRONOM version rather than to install the Service locally.

## Tools

Before describing some of the individual tools used in the scenarios above it is worth noting that in principle, the toolset described for scenarios 1 and 2 could be made available at any university – all the technologies exist and are robust, however the processes described rely on fairly complex interactions between the user and the university's ICT infrastructure which would be somewhat different on each campus. It is not an installation that is likely to be achieved through a simple installation CD.

The approach described in scenario 3 has been tested to the proof-of-concept stage and the tools necessary to carry it out are available.

### The iVia descriptive Metadata tool

The RepoMMan project team at the University of Hull spent some time searching for a tool that could do an adequate job of extracting descriptive metadata from a text file. The iVia tool is part of the Data Fountains software<sup>1</sup> developed at the University of California, Riverside (UCR). At the time (2005-2007) this was the only software that the team found that could take “any” text file and produce metadata of reasonable quality. The word “any” here is intended to indicate that the subject of the content is unknown and that the text cannot therefore be tested against a controlled vocabulary. The tool is not actually available from UCR as a Web Service but was adapted by the team at Hull.

The software (and thus the Web Service developed around it) analyses a Word or PDF file and attempts to extract a range of descriptive metadata from it. There are a considerable number of ‘switches’ for what might be extracted, but they include

- Document title
- Author(s) name(s)
- Abstract
- Library of Congress classification
- Library of Congress subject headings
- Language
- Keywords
- Acronyms
- Proper names and capitalised phrases

---

<sup>1</sup> Data Fountains. See: <http://datafountains.ucr.edu/> (Referenced 16 June 2009)

### JHOVE tool

The JHOVE (JSTOR/Harvard Object Validation Environment<sup>2</sup>) was a joint project of JSTOR and the Harvard University Library. It was developed as a tool for validating the format of a range of digital file types and checking that files are ‘well-formed’. In addition to doing this, however, it extracts a range of technical information that can be useful, for instance file size, images dimensions where applicable, and for digital photographs the camera EXIF information. This widely used tool is being developed into a new version, ‘JHOVE2’, by a consortium consisting of the California Digital Library, Portico and Stanford University.<sup>3</sup>

As with the iVia tool, JHOVE was not designed as a Web Service but was relatively easily ‘wrapped’ by the team at Hull to function as one.

### PRONOM and DROID

PRONOM has been developed by TNA as a resource, amongst other things, “to provide impartial and definitive information about file formats ... required to support long-term access to electronic records...”<sup>4</sup> The tool “measures those properties of the object which are significant to its long term preservation.”<sup>5</sup> The DROID tool is used in the scenario above to analyse a file and to provide the unique file signature(s) associated with it. Still within the publishing process, the DROID signature is then used to query TNA’s remote PRONOM service to retrieve information about the format identified by the signature(s).

The DROID tool is available to download as a standalone Java application, which also includes the Java APIs for software integration. The team at Hull made use of the APIs to wrap DROID as Web Service. The PRONOM tool is provided by the TNA as a REST based remote Web Service that can be integrated into local Ingest and other orchestrations.

### Local Web Services

In addition to the Web Services listed above the scenarios above rely on a further range of Web Services. As noted in the introduction, some of these are an integral part of the Fedora repository software. Other Services have been developed locally to perform specific jobs, some may be of general applicability others will be highly institution-specific. For instance, in all the scenarios above one item of descriptive metadata required will be the ‘creator’ of the content. In the first scenario the iVia tool might well extract this from the document text, in the second it is unlikely to be found in the JHOVE-derived metadata (although some digital cameras will allow a photographer’s name in the EXIF information they embed in a file). Clearly, one could rely on the creator to type it into the metadata form, but a more enlightened approach would be to try and find the name in the user’s computing environment: if (s)he is using the submission tool logged into a university portal the local

---

<sup>2</sup> JHOVE See: <http://hul.harvard.edu/jhove/> (Referenced 16 June 2009)

<sup>3</sup> JHOVE2 See <http://confluence.ucop.edu/display/JHOVE2Info/Home> (Referenced 25 June 2009)

<sup>4</sup> Brown, Adrian *The PRONOM Service: A technical registry to support long-term preservation* at [http://dlmforum.typepad.com/Paper\\_AdrianBrown\\_ThePRONOMService.pdf](http://dlmforum.typepad.com/Paper_AdrianBrown_ThePRONOMService.pdf). (16 June 2009)

<sup>5</sup> *ibid*

institution could deploy a bespoke Web Service that extracts the user name from the portal environment.

## Standards

A number of standards are referenced above. Further information can be found at the URLs noted below:

### Dublin Core (DC)

The Dublin Core Metadata Initiative has developed one of the most widely implemented metadata schemas.

See: <http://dublincore.org/> (Referenced 16 June 2009)

### UKETD\_DC

The UKETD\_DC Application profile has been developed by Cranfield University, in collaboration with partners in the development of EThOS, for the specific purpose of describing UK electronic dissertations and theses in a standard way. It consists of the basic DC elements and recommended qualifiers and domain specific extensions ('uketdterms').

See: [http://ethostoolkit.cranfield.ac.uk/tiki-index.php?page\\_ref\\_id=69](http://ethostoolkit.cranfield.ac.uk/tiki-index.php?page_ref_id=69) (Referenced 16 June 2009)

### Fedora Object XML (FOXML)

On the Fedora Commons wiki, FOXML is described as:

*A simple XML format that directly expresses the Fedora Digital Object Model... In addition, FOXML can be used for ingesting and exporting objects to and from Fedora repositories.*

See: <http://www.fedora-commons.org/confluence/display/FCR30/Introduction+to+FOXML> (Referenced 25 June 2009)

### BPEL

WSBPEL, often just BPEL, is an open standard maintained by Oasis. BPEL provides a comprehensive language for defining orchestrations of Web Service interactions. The most comprehensive review of BPEL is that offered by Wikipedia:

See: <http://en.wikipedia.org/wiki/BPEL> (Referenced 16 June 2009)

For the full WSBPEL (Version 2.0) Specification, see the Oasis reference material:

<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (Referenced 25 June 2009)

### Web Services: SOAP and REST

SOAP and REST are the two methods commonly used for implementing Web Services. Again, the most accessible summary of these is provided by Wikipedia. SOAP is a protocol recommended by the World Wide Web Consortium whilst REST is actually a style of software architecture which when implemented alongside HTTP can provide Web Services.

See: <http://en.wikipedia.org/wiki/SOAP> (Referenced 16 June 2009) and

[http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer) (Referenced 16 June 2009)

### Key Issues

One issue that is legitimately raised in the context of remote Web Services is “what happens when the remote Web Service is unavailable?” It should be noted that this could be simply a service failure (normal service will be resumed as soon as possible) or a more systemic problem (the service provider can no longer provide the service). Generally the first of these can, and should, be dealt with in the design of a Web Service orchestration by providing for an alternative source of the Service, retries, or for elegant management of the situation perhaps by rolling back part of the process; the second of them may have more drastic consequences. Inasmuch as it proves possible, users should choose their external Web Service providers with care, perhaps enter into appropriate Service Level Agreements, and ideally have contingency plans for dealing with the loss of a provider.

### Recommendations

#### Recommendation 1

The need for Web Services that can support quality metadata generation during repository ingest is not likely to decrease in the near future and it appears that such services are relatively few and far between. The JISC should continue to support work, as through the Information Environment Programme 2009,<sup>6</sup> which creates and/or identifies such Web Services and should ensure that they provide an easy means by which repository developers can review the list of known examples. (The same recommendation could be made in respect of post-ingest preservation Web Services but that is not the subject of this report.)

#### Recommendation 2

One of the Web Services detailed above is capable of extracting author names from a document, however identifying an author as, say, John Smith does not tell you *which* John Smith. A UK name authority would almost certainly be welcomed by the repositories community and a Web Service that could, from context, make an informed decision as to the likely John Smith identified would be a

---

<sup>6</sup> See: <http://www.jisc.ac.uk/whatwedo/programmes/inf11.aspx> (Referenced 25 June 2009)

bonus. The JISC is funding work in both these areas, for instance the Names Project at MIMAS in Manchester,<sup>7</sup> and should continue to develop it.

### Recommendation 3

The JISC should encourage repository software developers to make the use of automated metadata services feasible with their materials, either via means such as described here or through other means of embedding.

Richard Green  
IT Consultant  
'The Nook'  
Barton Street  
Barrow upon Humber  
DN19 7AR  
UK

r.green@hull.ac.uk  
or richard@thenook.karoo.co.uk

---

<sup>7</sup> The Names Project See <http://names.mimas.ac.uk/> (Referenced 25 June 2009)