Towards a Repository-enabled Scholar’s Workbench

RepoMMan, REMAP and Hydra

Richard Green
Consultant to the University of Hull IT Systems Group
<r.green@hull.ac.uk>

Chris Awre
Head of Information Management, University of Hull
<c.awre@hull.ac.uk>

This article was published in the May/June 2009 issue of D-Lib Magazine (http://www.dlib.org).

Abstract

Between April 2005 and March 2009, the e-Services Integration Group at the University of Hull undertook two Joint Information Systems Committee (JISC)-funded projects, RepoMMan and REMAP.

The RepoMMan tool developed a browser-based interface through which a user could interact with a private digital repository space to support the development of their works-in-progress. It went on to look at the processes involved in publishing such works to a public-facing repository and to investigate the possibility of generating metadata for the published object automatically.

The follow-on REMAP Project implemented the publishing process and also investigated how triggers might be embedded in the objects that were created which would help with management and possible preservation of the object over time.

The work of RepoMMan and REMAP has now been taken up in an international collaboration, the Hydra Project, which seeks to develop a repository-enabled “Scholars’ Workbench”. This will be a highly flexible system which will provide a search and discovery interface for a Fedora repository and which can be configured to provide interactive workflows around it for pre-publication development of materials and their post-publication management.

Introduction

It seems a long time ago that we contributed an ‘In Brief’ entry describing the work that the Joint Information Systems Committee (JISC)\(^1\)-funded RepoMMan Project (2005-2007) was to undertake.\(^2\) Nearly four years later, at the end of the follow-on REMAP Project (2007-2009), it seemed a good time to take stock of the work that we have undertaken, the lessons learned and our hopes for what promises to be an exciting future.

The RepoMMan Project set out to explore how a repository might be used to support the development of important works-in-progress by staff, and perhaps subsequently students, at a university. There were two ‘givens’ at the start of the project: that RepoMMan would use the
Fedora repository software\(^3\) and that the workflow behind it should be implemented by Web Services orchestrated using the Business Process Execution Language (BPEL)\(^4\). The project team did not believe that there should be arbitrary limits as to who might eventually use a repository in this way and thus it was that RepoMMan’s initial user needs survey covered members of the research, learning & teaching, and administrative communities. The project went on to develop a browser based tool that enabled a user to interact with a private repository space (‘My Repository’) and to use it as part of their workflow in developing materials. In addition, RepoMMan investigated how metadata for such materials might be generated automatically if and when a user decided to publish them to the repository proper.

The REMAP Project took the RepoMMan tool and implemented the publishing process, including the automated metadata generation. Additionally, in creating a new digital object for the public-facing repository, the REMAP tool embeds information that can be used for ongoing records management and digital preservation (RMDP). Using this embedded RMDP information the project has shown how, using a calendar server, alerts could be sent to appropriate repository staff over an extended period of time to ensure that materials in the repository are looked after effectively. In effect, the repository could become proactive in its own management.

The members of the team that developed RepoMMan and REMAP are now part of an international collaboration, the Hydra Project, which brings them together with teams from Stanford University, the University of Virginia and Fedora Commons. Hydra is developing an “end-to-end, flexible, extensible, workflow-driven application kit” for the Fedora repository software – in short, a configurable scholar’s workbench. The Hydra Project will incorporate many of the ideas developed in Hull over the last four years.

**The pre-requisites**

In our introduction we explained that the RepoMMan Project was started with two ‘givens’, that RepoMMan would use the Fedora repository software and that the workflow behind it should be implemented by Web Services orchestrated using the Business Process Execution Language (BPEL). It is worth taking a little space here to explain why these pre-requisites existed, not least because an understanding of the basics of Web Services and orchestration may help in understanding what follows.

At the time RepoMMan was started the University of Hull did not have a digital repository.\(^5\) Those who were developing the idea within the e-Services Integration Group (e-SIG) were intent on designing an institutional repository that could, potentially, hold all the University’s digital output. Thus it was that there was a need for repository software that was content agnostic because we did not want unduly to constrain the range of materials that it might eventually hold. Subsequent user needs analysis for RepoMMan confirmed our suspicion that the range was enormous. In addition we could only guess at the volume of content that might eventually accumulate and so we needed software with proven scalability. At the time we felt that, of the open-source contenders, only Fedora met these criteria adequately. Fedora also met a third criterion: it was Web Service enabled.

e-SIG were anxious to explore the possibilities of Web Services and their orchestration using BPEL. There had been some previous exploration of BPEL through the ASSIS project\(^6\) (a collaboration
between Hull and Stanford) to support the sequencing of assessments, enabling flexibility of testing according to responses received. The development of a RepoMMan tool provided an ideal opportunity to do this in a situation where we could see genuine advantages to the approach in a repository context. As the name suggests, Web Services are services that can be called over the web to do specific jobs for the user. The vague term ‘the web’ correctly implies that these services can, in principle, be located anywhere – within an institution or anywhere world-wide. There is something of an art to specifying Web Services at an appropriate level of granularity but the ideal is that each discrete service is useful in the context for which it was first designed but then, importantly, that it can be used to do the same job in other contexts without alteration. Thus it is that we have spoken of building a ‘Lego set’ of Web Services for the Hydra Project which will be discussed later. Services can be used and re-used in different contexts and sequences to assemble useful workflows.

Clearly there has to be a way of calling the Web Services in an appropriate sequence to construct these workflows. Additionally the workflows need to be able to support such needs as conditional branching and parallel tasks; BPEL is one approach for doing this and has the advantage of being a published open standard.

RepoMMan

e-SIG has a consistent record of placing real user needs at the heart of its development projects and thus it was that RepoMMan started with a period of extended user consultation. This work confirmed the team’s belief that there was clear potential use for a tool of the sort that was envisaged, with the following features:

- capable of handling almost any form of digital content
- permit safe storage
- allow access to the materials from anywhere they had internet access
- straightforward versioning of the materials over time
- easy to transfer content to a public-facing repository

As noted in the introduction, it was part of the contract with the JISC, the body that funded the work, that the tool would be implemented using BPEL-orchestrated web services over a Fedora repository. As work proceeded, it transpired that we were very much pioneers in bringing together this group of technologies. The use of Web Services allowed the tool to interact with other systems in the University, not just the repository, for a variety of purposes; indeed in the later REMAP tool we showed how this idea could be extended to use Web Services provided outside the University, specifically at The National Archives. Following early investigations into the range of BPEL servers available, RepoMMan used the open-source offering from Active Endpoints.

The RepoMMan Project ultimately produced a web browser-based tool for a user to manage their private repository space. The screenshot below actually shows version two of the tool as at the end of the REMAP Project but the basics remain the same:
The tool, written in Adobe FLEX, mimics an environment with which many users are already familiar: an FTP client. The left-hand side of the screen allows a user to browse their own computer to find files whilst the right-hand side represents the content of their private repository space using the analogy of a file structure on a disc. Digital objects are represented as ‘files’ and collections or groups of repository objects are represented as folders and can be manipulated in the ways that a user would expect. Files can be uploaded or downloaded as required. An additional feature that goes beyond the basic analogy is that double-clicking an object in the repository brings up a version history:

This is possible because multiple uploads of the same file create date- and time-stamped versions of it, they do not overwrite the original. Versions can themselves be downloaded or deleted.

Consider an example: when a user uploads a file to their repository space the FLEX interface initiates a sequence of Web Services orchestrated by the BPEL server. These manage, on the one hand, the copy of the user’s file to the repository file store – date- and time-stamping it in the
process, and on the other hand deciding whether a new digital object needs creating to hold the content or whether it is a version of existing content which should be updated. Once this has been determined the required process is actioned. In fact, something of the order of 15 different Web Services might be called upon to contribute to this upload process. Other actions within the browser client invoke different sequences of Web Services but each of these sequences will almost certainly include one or more of those used in the deposit process.

As noted in the introduction, the RepoMMMan Project also investigated the use of Web Services in the context of metadata generation but this is better dealt with in the context of the REMAP Project which looked at the process of publishing a user’s content into the public-facing repository.

REMAP

The REMAP Project took as its starting point the work that RepoMMMan had done. The basis of REMAP was to provide the functionality for a user to publish into the repository proper but, in doing so, add into the digital object thus created the potential for future RMDP work.

When a user selects one of their ‘files’ from the interface (really a digital object in their private repository space) the ‘publish’ button, shown greyed-out in figure 1, becomes active. If they click it they initiate the construction of a new digital object for use in the repository proper using, in effect, a wizard. The first thing that is requested from them is that they identify what the object is using a drop-down menu. They might say that the content is text and, by way of refinement, that it is a ‘thesis or dissertation’; or they might say that the content is an image and that the image is a ‘map’. The choice they make at this stage considerably influences the subsequent screens that the wizard displays to them during the process of gathering metadata. We strongly believe that the screens should be tailored to the content type and that as much as possible should be pre-filled for the user to accept or, if necessary, alter. In order to explain this approach more fully, consider in turn the two examples that we cited.

At Hull, the primary metadata for theses is held using the UKETD_DC schema (UK electronic dissertations and theses – Dublin Core). Accordingly the user will see a sequence of screens designed specifically to collect this information but, hopefully, with a lot of it pre-filled. Thus on the screen that asks for the title of the thesis, its subject and an abstract, we aim to pre-populate all the text boxes. In the background a series of Web Services are called which retrieve the content of the thesis from the file store and run it against the iVia descriptive metadata extraction tool developed by the University of California, Riverside. This analyses the document and attempts to extract these pieces of information. This is one of the metadata tools investigated during the earlier RepoMMMan work and in practice we find that it is remarkably effective; however if it ‘gets it wrong’ the user is at liberty to alter the information provided. The same approach of pre-filling what we can applies to subsequent pages in the process.

In the case of our image example the primary metadata held at present uses the Dublin Core schema. Thus the wizard will present a rather different set of screens to the user and again, Web Services are called to invoke appropriate metadata tools in order to offer information to the user where possible. In this case, as an example, the JHOVE tool developed at Harvard is used to provide file size and image dimensions as part of the metadata. The JHOVE tool is capable of
providing much useful information about many types of content and, in fact, for images we save the entire output as a potentially useful source of further information.

Once the metadata for a particular piece of content has been assembled, it is used in the construction of a new object for the repository proper. Again sequences of Web Services are called to copy the user’s original file and to build up the structure of the complete object according to a ‘content model’ that is applied consistently to all similar objects. This use of consistent and predictable structures aids the management of the object in the longer term. During this construction, new representations of the author’s material may be generated. A thesis submitted as a Microsoft Word file would be converted to pdf (the repository preferred format) automatically and the Word version saved as an archive copy. The image file would have a number of derivatives of different dimensions generated, each suited to a different purpose. The new object (and its content) belongs to the repository rather than the author which is to say that the author cannot alter its content. In the first instance the new object is placed into an accession queue where repository staff can check it before releasing it for exposure.

The processes described above effectively provide the publishing functionality that RepoMMan envisaged. However, the REMAP Project has enriched the publication process in order to better enable subsequent management and possible preservation of the content.

In setting up what could ultimately be a very big institutional repository we at Hull recognised that it would require careful and regular management into the future. Potentially there is a lot of work in this, remembering to do such and such every six months, or that there is a thesis to take out embargo on such a date, or… In addition, each repository task potentially calls for a number of routine side-tasks such as sending progress updates to an author publishing a document. How much better if the repository could become a player in its own management? The REMAP team investigated how a set of triggers could be embedded in an object at the time of its creation and then used, in conjunction with a calendar server, to initiate automated tasks or to alert an appropriate human to the need to carry them out. This last piece of work has been taken to the demonstration stage within the timeframe of the funded project but will require further development before being used in a production context.

Consider three examples which illustrate the sort of processes that might be involved:

If we follow further the thesis publication outlined above it will be useful, when the thesis becomes available in the repository, to send to the author an e-mail stating this and giving the URL that the thesis (and/or its splash page) has in the repository. This is achieved by building into the new object when it is constructed a trigger which will initiate the process without the repository manager being involved. When the object is released from the accession queue a ‘publication date’ is injected into its RMDP and with it a ‘to-do’ instruction that the author should be notified on or immediately after that date. Any such instructions are copied to the calendar server. Periodically the calendar server is polled for any unprocessed ‘to-do’ instructions for the day’s date or earlier and these are initiated using orchestrated Web Services. When the mail is successfully sent the ‘to-do’ instructions in the object and on the calendar server are flagged complete and thus an audit record starts to build up within the object itself. In line with e-SIG’s policy of using standards, and especially open standards, where possible, these triggers and calendar entries are implemented using a small group of calendaring standards, CalDAV, iCAL and xCAL.
In another situation it may be that a Departmental Secretary has published a set of past undergraduate examination papers to aid in future revision work. Clearly with time these become less relevant and so the digital objects for such papers will contain triggers which, after five years, notify the secretary that these papers should perhaps be hidden from students and be available only to staff. We anticipate that the secretary will be able to respond to such notification selectively if required, “hide these but not those”, and that the papers might be hidden automatically if no response is received within a specified time frame.

These first two examples have effectively been records management; consider finally something from the preservation perspective. This example also serves to demonstrate the use of remote Web Services. When each object is published into the repository proper we can run it against a local installation of the DROID tool\textsuperscript{11} developed by The National Archives (TNA) in the UK. This tool analyses the file containing the author’s material and attempts to identify exactly the file format and return a ‘format signature’. This signature can then be passed to TNA’s PRONOM service,\textsuperscript{12} which in turn returns information about the file format including any risks and recommendations. We have installed the DROID tool locally at Hull to avoid passing potentially large files to a remote service, however we then pass the DROID format signature to the remote PRONOM service at TNA. The outputs from both services are stored in the repository object with a date. A trigger is set to re-run the process at periodic intervals so that the repository manager can be alerted to any future risks that may, for instance, indicate that a format migration might be desirable.

It is a matter of regret to the REMAP team that we were not able to develop this RMDP work as far as we would have wished during the time of the funded project. However, there are often pitfalls lurking when a team embarks on pioneering work with open-source software and this proved to be the case here. We found that some of the software implementing our chosen standards was not as mature as we had been led to believe and this greatly slowed our development work. Nevertheless we have done enough to demonstrate to our satisfaction that there is considerable potential in this approach and that we wish to take it further.

**Hydra**

At the Open Repositories 2008 conference held at Southampton University in April of that year, the RepoMMAn/REMAP team was approached on behalf of the University of Virginia who were keen to take the work of the two projects and to build on it. It was agreed that a meeting should take place in the early autumn to discuss the idea. The meeting took place in September 2008 and Hull and Virginia were joined by Stanford University and Fedora Commons, this last the organisation set up to further develop the Fedora software and related technologies. Much was discussed and the outcome was a joint commitment to develop, over a period of three years, a Fedora application kit which would not only serve the needs of the partners but which could be used by others. This is the Hydra Project.

The universities share a view that a repository can potentially be used at all stages in the life-cycle of almost any born-digital material: stages from development, through exposure on the web, to long-term management and preservation. Hydra will provide, within a reusable application framework, a ‘Lego set’ of web services and templates that can be configured and reconfigured to suit a wide
range of different workflows that institutions might have or might develop to manage their digital content.

Hydra will have configurable interfaces to help authors and creators develop their content, perhaps in collaboration with others, expose it on the web, and for the institution’s repository managers to manage and possibly preserve it long-term. Hydra will also have a search and discovery interface through which users can explore appropriate areas of the host institution’s repository.

Recognising that not all potential users will wish to work with BPEL, each of the three universities is developing a separate approach to orchestrating their Web Services. At Hull we will continue to work with BPEL whilst Stanford and Virginia will each develop a different approach. The search and discovery interface will be written using Ruby and will incorporate Virginia’s ‘Blacklight’ search tool.

Work is well under way and the partners each hope to have the core of a production system in place for the academic year 2009/10. The software will then be successively generalised and made available to others; in addition the toolkit will be developed for other users to implement RepoMMan-like processes prior to repository submission and REMAP-like processes following deposit, each according to local needs. This end-to-end process has been termed the “scholars’ workbench” and is an approach that we involved in Hydra are not alone in developing. At the Open Repositories 2009 conference Fedora Commons is to launch a ‘solution community’ to bring together all those with similar interests. (The Scholar’s Workbench Solution Community will be one of a number of such solution communities that Fedora Commons is fostering.)

More information about the Hydra Project can be found on its website at: <https://fedora-commons.org/confluence/display/hydra/The+Hydra+Project>

More information about the Scholar’s Workbench Solution Community can be found at: <https://fedora-commons.org/confluence/display/FCCWG/Home>

Acknowledgements
Over the period of RepoMMan and REMAP many people, from the UK, continental Europe, Australia and the US, have contributed directly and indirectly to the work. It would be impossible to name them all but they have our thanks; that said, two groups should be mentioned by name:

We acknowledge with gratitude the contributions of the Joint Information Systems Committee (the JISC) in funding the RepoMMan and REMAP Projects, and in contributing towards the travel costs associated with getting the Hydra Project under way.

We acknowledge, too, the significant contributions made to REMAP and the thinking around it by our good friends in Spoken Word Services at Glasgow Caledonian University (David Donald, Caroline Noakes, Iain Wallace and, for the first year, Graeme West) who were partners in the project.
References

1 See: <http://www.jisc.ac.uk>


3 The Fedora website can be found at <http://fedora-commons.org>

4 Business Process Execution Language (BPEL), see <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpe>

5 Hull’s beta institutional repository can now be found at <http://edocs.hull.ac.uk>

6 ASSIS project, see <http://www.hull.ac.uk/esig/assis.html>

7 The Active Endpoints software is now called ActiveVOS. See: <http://www.activevos.com/>

8 See: <http://ethostoolkit.cranfield.ac.uk/tiki-index.php?page_ref_id=47>

9 The iVia tool is part of the larger Data Fountains software initiative. See: <http://datafountains.ucr.edu/>

10 See: <http://hul.harvard.edu/jhove/>
