



# RepoMMan Project

---

D-D8

Experiences with Fedora during the project's first year

Richard Green

July 2006



## The RepoMMan Project

<b>Project Director:</b>	Ian Dolphin, Head of e-Strategy, University of Hull (i.dolphin@hull.ac.uk)
<b>Project Manager:</b>	Richard Green (r.green@hull.ac.uk)
<b>Technical Lead:</b>	Robert Sherratt (r.sherratt@hull.ac.uk)
<b>Repository Domain Specialist:</b>	Chris Awre (c.awre@hull.ac.uk)

The Repository Metadata and Management Project (RepoMMan) at the University of Hull is funded by the JISC Digital Repositories Programme. The project is being carried out by the University's e-Services Integration Group (e-SIG) within Academic Services.

## Introduction

The RepoMMan project is researching and developing a workflow tool to facilitate the use of a Fedora-based Institutional Repository at the University of Hull. At the time the project was proposed, Fedora was little used in UK academia and it was felt that an account of the team's experiences with the product may be useful to those that follow. This is that account.

During the first few months working with Fedora it became clear that the documentation provided by the Fedora team was not wholly sufficient for users coming new to the product. Accordingly, we at RepoMMan decided to repurpose one of our deliverables, 'D-D4 Iterative development of Fedora materials' to become a beginner's guide to the product which could be offered, without warranty, to the Fedora community. That document might usefully be read alongside the present report.

Our 'experiences with Fedora' related here are focussed primarily on experiences with the software. However, it would be wrong to exclude completely our experience of interacting with the Fedora community; this was an essential part of our learning experience. We have therefore touched on it in this document though not in the detail that has been afforded to our development work.

Work with Fedora developed along two separate but parallel routes: Richard Green undertook the research elements of the project and most of the work with the repository *per se*, whilst Simon Lamb undertook the practical, development work that will lead directly to the production of RepoMMan's workflow tool. These aspects will, initially, be dealt with separately.

## Part I: Working with Fedora as a repository.

### 'Out of the box,' onto XP®: autumn 2005

The Fedora repository software and all the associated documentation is to be found on their web site at:

<http://www.fedora.info/>

The 'download' page provides access to a binary-only copy of Fedora, or to a source distribution consisting of the source code and libraries. In each case there is a Windows or a Unix version available. Since the release of Fedora 2.1, there are also source and binary downloads for a number of additional services and tools. There is a software prerequisite that Fedora's host machine must be running Sun's *Java Software Development Kit v1.4.2* or later.

When we started work with Fedora in the autumn of 2005, the product was at version 2.0. For early experimental work we used the Windows binary version on a Windows XP (SP2) box.

In addition to the software, Fedora make available a range of documentation and tutorials. The use of the word 'range' is deliberate. There is no single link that can be used to print or download the complete set of documentation; rather the user has to follow a considerable number of links and to print the components one at a time. Nor, in autumn 2005, was it a simple case of working down a page with a full list of links: over the course of the year we stumbled across a small number of items that were referenced only from within other pieces of documentation. This particular experience we found frustrating. With the release of Fedora 2.1, the 'documentation' area of the Fedora site did acquire a single page with an almost complete list of links.

Once printed, the totality of the documentation is quite comprehensive. However, we came across places where its authors seemed to assume too much knowledge on the part of a new user, where quite complex ideas were dealt with somewhat summarily, or small gaps where important (to us) concepts were not dealt with at all. It was these perceived deficiencies that led us to produce the 'Beginners Guide' referred to in the introduction.

The downloaded Fedora product includes a range of demonstration objects and a range of disseminators that can be used with them. It seemed to us a sensible approach to analyse these and to develop our own first objects and disseminators using them as models.

This and earlier versions of Fedora had only limited security which allowed protection of management functions but no protection against retrieval of the 'digital payload' from repository objects. (Note that much improved security became available with Fedora version 2.1)

We had no great difficulty in getting Fedora 2.0 to work across a peer-to-peer network but were somewhat concerned that its response times were generally measured in seconds rather than milliseconds. This did not overly concern us because it did not seem to be an issue on the very active Fedora-users' mailing list and we felt sure that it would have been if, indeed, this was a known issue. (In the event, we suffered from this sluggish response until we were able, some months later, to install Fedora on a dedicated Solaris-based server at which point the problem vanished.) The most complex part of the installation process was found to be setting up, which is to say editing, Fedora's configuration file. Whilst the documentation provided for this part of the installation process is perfectly adequate, the file is quite large and the process needs a lot of care. For one coming to it as a Fedora novice, the task can seem rather daunting.

The Fedora-user mailing list mentioned in the last paragraph should be seen as an essential element of a Fedora installation! Membership is free and it is an invaluable forum where users

of the product world-wide can share experience and interact with the Fedora development team. (In 2006 a Fedora wiki was launched to complement the mailing list.)

The software as downloaded comes with the McKoi pure Java database, but it is possible easily to use MySQL or Oracle 9i; other databases are possible but less straightforward. This database is used to track Fedora objects and their properties. The Fedora 'product' also has a version of Tomcat inbuilt to provide a web front-end. As at Fedora 2.0 and 2.1 it was not easily possible to run Fedora under a different instance of Tomcat. The Fedora configuration file sets up a range of parameters including the type of database that will be used by Fedora for object tracking. Other parameters include the 'master' username and password, the IP address and ports that Fedora will use, the namespace that the installation will use for object persistent identifiers, the IP address(es) from which management calls will be allowed (this was replaced in Fedora 2.1 by external security control), and so on. In addition to the database mentioned above, Fedora has an integrated Kowari datastore to provide a searchable 'graph' of relationships between objects, the so-called resource index. The Kowari datastore is not easily amenable to being replaced.

Once installed and running, potential users and managers have access to Fedora through a web browser, which will provide a straightforward search and retrieve function, and through an administrative client which uses web services to provide remote management functionality. A number of utilities are provided on the server for more 'under the bonnet' management. Amongst these utilities is one that deserves more attention than it perhaps gets. A tool called 'fedora-rebuild' allows one to rebuild Fedora's resource index, on which relationship searching is based, or the database which tracks Fedora objects and their properties if either becomes corrupted. What is perhaps not well explained is that it can be used to rebuild *both*. Thus if both the resource index and the object database are destroyed all is not lost; Fedora can rebuild both by crawling its object store. This is an extremely powerful and, in our opinion, under-reported feature.

Once running, it was possible to explore Fedora and to start to understand how the various aspects of the repository interact. The first surprise for the novice is perhaps that 'everything is a Fedora object'. The notion of a digital object should not be unfamiliar to anyone who has got this far with a repository, however that fact that a collection of digital objects is itself represented by a Fedora digital object may seem strange at first sight. Even more strange, the fact that the components of the disseminators that control how an object might be distributed and presented are themselves Fedora objects.

A basic Fedora object consists of a number of components, two of which are compulsory.

Firstly, as an object is created it is associated with a 'pid'; a persistent identifier. This can be machine assigned or user assigned but once used can never be duplicated. The pid consists of two parts: a prefix identifying the origin of the object (in our case we simply used "hull:") and a unique identifier for the object itself. Fedora assigns numbers, humans can assign text. Thus valid pids would be hull:12345 and hull:barcelonaImageCollection. A pid is compulsory.

Within the object will be a range of 'datastreams'. These refer to the various elements that make up the digital 'meat' of the object. One of these is compulsory: a datastream, labelled 'DC', that contains Dublin Core metadata about the object. The dc:identifier and dc:title elements are required and created with the object. The dc:identifier is the object pid, whilst the dc:title is user assigned.

A 'real' Fedora object will then have at least one datastream which provides a data source. Each datastream can either provide a source of XML or of data that conforms to a mime type (images, text, data etc). In either case the content management can take place within the repository or external to it; in the latter case it will be referenced by a URL (not, note, a file path). Each datastream must be assigned to one of four categories, or control groups.

- managed content: the content is managed within the repository

- inline XML: a special case of the above in which the XML becomes part of the Fedora object's structure
- externally referenced: content held in a datastore outside the repository and referenced by URL, and
- redirected content: externally referenced content which is simply fetched, unmediated, by Fedora on request. This is useful for a range of material including websites with relative links and streamed media

A Fedora object may have multiple datastreams referring to content - perhaps the same document in different formats, or the same image in different sizes.

It is likely that a user will also create one or more datastreams containing metadata about the content of the object and possibly a datastream establishing structural relationships between this object and others in the repository. There are many possibilities. It should be noted at this point that the compulsory 'DC' datastream mentioned above is intended for the purposes of internal management, it is not intended to form a datastream on which external discovery searches would be based; that functionality would be provided by one or more additional metadata datastreams.

If and when datastreams are updated, Fedora stores a new version keeping the older one(s) accessible.

In addition to datastreams, a Fedora object may contain one or more 'disseminators'. Disseminators associate the data in a digital object with web services which provide some form of dynamic delivery to the user. Thus, in a simple case, an Fedora object might point to an image in TIFF format, but an associated disseminator could transform it and deliver it to the end-user in JPG format. An object representing a collection of texts may have a disseminator that dynamically queries the repository to find the members of that collection, thus obviating the need for a static list which requires update when a new collection member is created.

Thus a typical Fedora object might be represented as follows:

Persistent Identifier (PID)		Unique identifier
.....		
Disseminators	}	Disseminators to control exposure of internal metadata and other datastreams
.....		
Structural metadata		
Object metadata		
.....	}	Protected content making up the basis of the digital object
Datastreams		

## First objects

With an eye to a potential demonstrator, we started out by creating image objects. These were based on the objects in Fedora's demonstration 'Smiley Collection'. The simple image objects were straightforward to generate using the Fedora Admin client and, by effectively cloning the disseminators provided for the Smiley Collection, we were able to make the objects behave as what we now know to be an implicit collection.

This, though, exposes the first shortcoming that we identified in the Fedora documentation: although much is made of the product's ability to handle collections of objects, the mechanisms for doing this seem nowhere to be adequately explained. As things worked out, it was a private e-mail from one of the Fedora team that gave us a clear explanation of the processes involved and the differences between explicit and implicit collections. The

collections depend on asserting relationships between the objects. In an explicit collection, the collection object asserts that, for instance, it 'has members' this, that and the other; thus the collection object has a complete list of its children. In an implicit collection, the collection object knows nothing of its children other than how to query the resource index to find them; each child asserts that it is a 'member of' the collection object. Relationships within Fedora are potentially many-to-many so that a particular digital object may, in principle, form part of a number of collections or none.

As noted above, the objects were created using the Fedora administration client. We found that this worked perfectly well and that the documentation for its basic use was adequate. The quibbles we had with its documentation at this stage were to do with the fact that it was sometimes not clear how fields that you filled in on-screen related to the finished object. For instance, how was one supposed to know that the blank field called 'label' that appeared when creating an object became its title? Or that the 'datastream ID' became the datastream's label. All sensible enough and very quickly understood - but not spelt out in the documentation.

In this version of Fedora (2.0) the management functions made available through the administration client required authorised access; discovery and retrieval of object contents through Fedora's web interface did not.

Handling these initial objects as an implicit collection required the provision of a query to the Fedora resource index. Initially this was developed using a Fedora-provided query as a template but, again, the Fedora documentation does not really cover the writing of queries adequately. Our initial work in this area involved a great deal of trial and error. Fedora's resource index is derived by default from a Kowari database and some time later we discovered a tutorial on the Kowari website<sup>1</sup> which greatly increased our understanding of the query language and its syntax.

In all, it took about two weeks to become adequately familiar with objects and disseminators and how these might operate together to form and display collections of fairly rudimentary objects. These early objects contained only minimal metadata (sufficient to support the collection methodology) and whilst, in retrospect, they conformed to an implicit content model this was more by accident than design; certainly it was not the result of any planning process. (A content model would define, for instance, the number and type of datastreams and disseminators in a conformant object.) This 'adequate' familiarity should not be taken to mean a clear understanding of disseminators, in particular!

Fedora objects can store their 'digital payload' internally or can refer to an external location where it is to be found and retrieved (there is also the possibility of Fedora simply redirecting a web resource to the user). In view of the way that we anticipated the repository developing at the University of Hull we decided that external digital payloads would be our norm. At a meeting hosted by the Open University, it was suggested that in any repository externally held data files should be named in such a way that their true location in the repository structure was obvious. To explain: although management of a repository should normally be through a controlled interface, there will be occasions when, say, administrators deal directly with the file structure on the host computer(s). Sooner or later it is almost inevitable that a file will get moved from its rightful position and become 'lost' to the repository. The argument runs that, if its filename clearly identifies where it should be in the repository file structure, it can be searched for on the machine, (hopefully) located and put back where it belongs. As long time users of computers, this scenario seemed familiar to us and we decided that Hull's repository should adopt this way of working.

The process of working through the creation of objects and disseminators was a useful one - not just in terms of coming to grips with Fedora, but also in terms of thinking about the ways in which the aspects of Fedora objects that one was exploring might relate to their eventual use in an institutional repository at Hull.

---

<sup>1</sup> <http://kowari.org/oldsite/187.htm>

As has been noted elsewhere<sup>2</sup>, Hull's view of a repository is a very broad one and working with collections of objects led one to consider how collections might form the basis of a private repository area. It became clear that there would be ways of mapping Fedora's 'collection' mechanisms onto a file structure metaphor such that collections could masquerade as directories. At the same time, working with Fedora's compulsory Dublin Core (DC) datastream started one thinking about the metadata that a repository object would eventually need and how this might be mapped onto the Dublin Core fields. It was some considerable time later that an e-mail on the Fedora-users list made it clear that the inherent DC datastream was intended solely for Fedora's internal use and not as the basis for external searches. This raised the issue that Fedora 2.0 did not provide a mechanism for searches based on a different metadata datastream.

## First content models

Some three working weeks, or so, into our experimentation we were beginning to understand how Fedora objects and disseminators would work together for us, and to understand that we needed to define a set of content models if we were to deal with objects in a structured, logical fashion. In essence, a content model is simply a definition of a conformant Fedora object in terms of the number and type of datastreams and disseminators.

Initially, we thought to define a 'standard colour image model' and a 'standard public text' model each of which would be immediately applicable to our work in developing demonstration materials. We were later to understand that content models also gave us a means of exercising security within the repository (although we had to wait for the release of Fedora 2.1 to realise this aspect of the process). The development of content models also made us think more deeply about the type and range of metadata that would be appropriate for an enterprise service in Hull. Our thinking at this stage was to use Dublin Core (DC) and qualified Dublin Core as the main block of metadata (to be distinguished from a Fedora object's internal DC metadata), supplemented by elements from MODS<sup>3</sup> and MIX<sup>4</sup> as appropriate. We also found ourselves thinking about the basic object-related services that might be provided for potential users through Fedora disseminators.

Disseminators is one aspect of Fedora which does pose some issues. Disseminators are made up of two parts: a so-called BDef object which defines the method calls associated with a particular disseminator, and a BMech which implements them. It is not possible to replace a BMech that is bound to one or more digital objects. Thus, if an error is found in part of a BMech, or if it is desired to extend the functionality of a BDef/BMech pair, it is first necessary to disassociate the BMech from all objects that subscribe to it. This makes for a difficult process. Nor should it be assumed that the need is uncommon; BMechs are difficult objects to construct, even given the tool in the Fedora Admin Client, and mistakes - sometimes undetected at first - can creep in. The problem of 'locked' BMechs is one that has exercised the Fedora community and to which an alternative approach is being considered; this new approach is described later in the document.

Our personal feeling is that the whole area of disseminators in general, and the construction of BMechs in particular, is one that would benefit from much more by way of documentation and tutorial material.

The draft content models that we developed at this stage are attached as Appendix 1 and Appendix 2. They are perhaps of some interest in that they expose our earliest thinking about metadata and show those areas of it where we felt reasonably confident and those where we were still 'feeling our way'. The pair are attached so that the reader can compare and contrast the needs of the two different types of object. As will be seen later, it was at the very end of the RepoMMan project's first year that we started to entertain doubts about our initial approach to metadata.

---

<sup>2</sup> See, for instance, Green (2006) p

<sup>3</sup> MODS <http://www.loc.gov/standards/mods/>

<sup>4</sup> MIX <http://www.loc.gov/standards/mix/>

## **Fedora Users' Conference, National Library of Wales**

Late October saw a meeting at the National Library of Wales, 'Digital Asset Management using Fedora'. This was a very useful conference for two main reasons. Firstly, it was the first conference at which RepoMMan had been asked to speak and thus it helped us to crystallise some ideas as we developed the presentation. Secondly, it was our first contact with a number of Fedora users from the UK and elsewhere. The contacts established with National Library staff have been useful over a long period, and the chance to meet Thornton Staples, one of Fedora's co-Directors, was opportune.

## **Fedora workflow group**

Towards the end of 2005, the Fedora team announced the formation of a 'Workflow Working Group' to look at the development of a workflow tool. The team were well aware of Hull's involvement in this field and we were invited to be part of the group.

Over a period of time some preliminary work was done in the US to determine how best a workflow tool might be developed. The end result was a decision that Fedora's work might best be developed using JBPM (Java Business Process Management) leaving Hull to follow the BPEL-based development funded by JISC as the RepoMMan project. These were seen as potentially complementary developments.

## **Fedora 2.1: February 2006**

At this stage in our development work Fedora 2.1 became available and was installed on our testbed machines. From our point of view the major new feature in 2.1 was the addition of XACML security to the product. Security was an issue to which we had given a lot of thought, but little development time, preferring to wait until it could be managed largely from within Fedora rather than from without.

The upgrade process from Fedora 2.0 was relatively straightforward. (Note: we actually had some problems installing Fedora 2.1 on one Windows XP machine, but on most there was no problem. This particular issue was replicated at a very small number of sites around the globe.) The major change to previous configuration options was that the internal resource index had to be 'turned on' (a setting in the configuration file) whereas in 2.0 that had been the default setting. New to Fedora 2.1 was the need to declare an initial security model. With an eye to our future needs, we opted for 'no-ssl-authenticate-all'. This setting causes Fedora to protect all its management functions, as with 2.0, but additionally requires authentication for all access calls.

Our preliminary work with the XACML security in Fedora uncovered a number of major problems with the release. As we experimented with the demonstration policies provided and tried adapting them to our own situation we discovered a number of situations in which Fedora denied all access to objects - even to the Fedora Administrator. We reported these problems to the Fedora team using the Fedora mailing list and over a period of a few days two distinct problems were uncovered and fixed: some of the demonstration security scripts were malformed, and - more fundamentally - a bug was discovered in one of Fedora's Java modules that caused it to refuse all authorisations when certain kinds of security script were used in combination.

In terms of our 'experience with Fedora', accepting that these problems should perhaps have been discovered before the release of 2.1, the interaction with the Fedora developers was a very positive one and we always felt that they were working to solve our problems quickly.

## XACML security

Once these initial problems with security had been sorted out we set to work exploring what was and was not possible. Initially we took the approach that we had taken with objects, that of tinkering with demonstration material to see how it worked. However, we soon gained familiarity with the basics and started experimenting on our own.

Fedora security can be attached at almost any level in the Fedora hierarchy of operations. It can be attached to complete objects or complete disseminators, or it can be attached to elements within an object (eg a datastream) or a disseminator (a method), or it can be attached to specific Fedora service calls. In addition to properties of the repository and its contents, the security system can access such things as 'roles', either from a Tomcat user list or an LDAP server, or the user's login name. All this makes for a very flexible security system though one that could get inordinately complex if not managed sensibly.

After due consideration it seemed to the development team that security in the repository might best be managed by assigning all objects a content model and applying our primary security measures to these models. In other words, all objects conforming to a particular object model would automatically inherit the security assigned to that model. This approach limits the number of security scripts that have to be managed to not many more than the number of content models that the repository supports. Objects in the repository private space, where it would be inappropriate to impose a genuine content model, would be managed by assigning them to a pseudo-model which implies no structure but which can be the subject of security restrictions.

One of the demonstration security scripts provided with the Fedora 2.1 download demonstrated how repository users could be denied direct access to all datastreams in the repository except through an approved disseminator (ie one that related to the particular object in question and for which they had the necessary security level). This idea appealed to us as a primary mechanism for limiting the repository's vulnerability to hacking and it will be implemented right across the public area of the repository. As noted earlier, Hull will store its digital payloads outside the direct control of Fedora and, here, IP restrictions will be placed on the directory tree to prevent direct access by unauthorised users; indeed we foresee that the only IP address that the tree will need to accept will be that of the Fedora server itself given that all the major components of Hull's repository services will be co-hosted on the same box.

## A dedicated server: March 2006

Late March of 2006 saw the delivery of a dedicated server for our Fedora testbed. This was a Solaris-based twin 2.6GHz processor machine from Sun Microsystems with 4Gb of RAM and 73Gb hard disk space. As noted in the introduction to this document, commissioning of the new server confirmed our belief that the performance issues that we were seeing would vanish as soon as an appropriately powerful server was provided.

With the coming of a 'fast' implementation it became sensible to produce a number of small 'demonstrators' to show interested parties. As noted earlier, the RepoMMan team shared a view that image objects would make a good first collection.

Bringing together all our previous experience we developed an implicit collection of images from the University Photographic Service and applied to them what we considered a reasonable security model.

The image objects conformed to the (then) current version of the Standard Colour Image Model. They contained four image datastreams (archive, fullsize, screensize and thumbnail), DC metadata and EXIF metadata. Direct access to object datastreams was forbidden to all but administrative users, but the screensize and thumbnail images could be accessed by staff and students of the University - as could the DC metadata. To access the fullsize and technical (EXIF) metadata one had to be a member of the University Photographic Service or of the

Marketing and Communications team. (Our security model always allows Administrator access to all Fedora objects.) The collection thumbnails were gathered together and displayed on a web page by repository tools. Against the thumbnails were links to obtain larger versions of the images, or the metadata, but these were appropriate to the user's role. Staff and students were not given links to the higher order images or metadata.

A document collection (actually of RepoMMan project documents) was developed in a similar fashion. In this case security mandated that the pdf version of the documents could be accessed without a password but using an appropriate disseminator.

A chance remark at a project meeting along the lines of "wouldn't it be nice if you could click on an image and see on a map where it was taken" led to a second image collection being developed. The images were of sites around the world and included in their metadata a latitude and a longitude field. A disseminator was developed that could parse out this information and pass it to the Google Map API, throwing up a window with a zoomable, aerial view and map of the camera location. This collection was set up as a 'private' collection accessible only to Fedora administrators and to the 'owner' of the objects, this last to be determined by comparing the object's owner ID with the login ID of the user trying to access them. ("to be determined" because Fedora does not yet implement an ownerId property, we anticipate its release in version 2.2; in the meantime the process has been 'fudged'.)

These collections were used in a number of meetings, internal to the University, to demonstrate possibilities.

## **The Fedora Wiki**

The spring of 2006 saw the launch of a Fedora wiki<sup>5</sup> to complement the facilities provided by the Fedora mailing list. Whilst not of immediate relevance to a discussion of our 'experiences with Fedora', the wiki - as it developed - became a place where Fedora users could post non-immediate issues about the software, including a feature 'wish-list', a section that we initiated and used. In addition, the wiki became a place where experience could be shared amongst users by making available anything from snippets of ideas, through short bits of code (for instance XACML), to pieces of documentation. We, at Hull, take advantage of the wiki to learn from others and to share the products of our own experience.

## **Fedora UK & Ireland User Group**

At the Fedora conference at the National Library of Wales, it had been suggested that a UK User Group be set up and the representatives from the University of Hull had agreed to set this up.

In the spring of 2006 we contacted all the known users of Fedora in the UK. Through these initial contacts, others were discovered and a user in Ireland asked if he could take part; the group became the UK and Ireland group. It met for the first time in May and, during the first part of the meeting, delegates each spent ten minutes describing their use of the software. The second part of the meeting discussed whether a UK&I group was a useful concept and what it might do. It was decided that a group was needed and that it should meet again in October.

We, at Hull, made a number of useful contacts that day and, in particular, discussion with Martin Dow of Rightscom proved especially fruitful; this is covered in Part II of this document.

Appendix 3 details the membership of the group as at the end of June 2006

---

<sup>5</sup> [http://www.fedora.info/wiki/index.php/Main\\_Page](http://www.fedora.info/wiki/index.php/Main_Page)

### **Fedora 2.1.1: May 2006**

In late May, we upgraded to Fedora 2.1.1. In truth, this version of Fedora had been available for some weeks but initially we saw no need to adopt it. 2.1.1 fixed a number of ingest performance issues and some of the security bugs which we had been instrumental in identifying. (The security fixes had previously been available only as patches for download; 2.1.1 supported them as native.) It was work in the web services area that required us to switch and this is dealt with in the second part of this document. Sufficient to note here that the upgrade process was fairly painless although there were some gaps in the documentation (such as the need to change some environment variables on the server to reflect the new directory paths).

Almost a year into our 'Fedora experience', we were thinking more and more about the 'service' aspect of what we were developing. Although not strictly part of the RepoMMan project, we were beginning to have concerns about Fedora's current lack of a user-oriented search tool which could, for instance, perform full-text indexing on Fedora objects. We were also somewhat exercised about the plethora of possible metadata formats that we might have use for. In addition, we were still looking for a text metadata extraction tool that could potentially form part of the RepoMMan workflow tool's submission process. The Fedora Users' conference provided a forum in which all these concerns were explored.

### **Fedora conference 19-20 June 2006**

In June 2006, two of the RepoMMan team attended the Fedora Users' conference at the University of Virginia in Charlottesville, US. This turned out to be an important part of our 'first year experience' and one that we would urge others to consider when subsequent conferences are held.

The conference proper lasted 1½ days and, apart from two plenary sessions, was divided into two 'tracks' that might loosely be described as 'technical' and more general. Between the two attendees from Hull, Richard Green and Chris Awre, all the sessions were covered. In addition, there was a meeting of the Fedora Workflow Workgroup on the day preceding the conference, a set of 'BOF' sessions on the first full day, and we (the Hull representatives) arranged a private meeting with University of Virginia staff on the day following the conference (to which came also some representatives from Yale).

Whilst a full discussion of the conference is not appropriate to this paper, a number of aspects relate to our 'first year experience'.

### **Workflow working group**

The meeting of the workflow working group revisited the group's charter in light of the fact that circumstances had conspired to prevent any development work being done in the period since the charter was drawn up. There was lengthy discussion of the means now available to produce a workflow tool and the considerable overlap between the interests of this group and the parallel Fedora 'preservation working group'. Eventually a consensus was reached that the group should probably look to integrate into Fedora a workflow engine on top of which units of functionality, developed by the preservation group, could be chained together. The engine might be based on BPEL, for heavyweight workflows, JBPM for less demanding situations, or simple messaging for simple workflows - a sort of workflow-lite. The totality of this work is seen to be such that the work would probably need to be grant-funded.

## Elements of the conference

Chris Awre is developing a separate report to JISC on the conference proper<sup>6</sup> and so we shall not attempt to replicate that here; rather to raise a number of issues that relate to previous comments made in this document and to potential developments at the University of Hull.

We have commented on the lack of a suitable search engine for use as a discovery tool by repository visitors. During the late spring and early summer of 2006, Gert Schmeltz Pedersen, at the Technical University of Denmark, had been working on such a tool (which by the time of the conference was available as a beta release). The tool takes either Lucene or Zebra as a plug-in and provides full-text searching of Fedora content: metadata datastreams and/or textual content of either managed or external data content. At the time of writing, the full-text indexing of files was possible for only .txt, .pdf and .html types, but he anticipated that more would follow. The search service, which is likely to be released as part of Fedora 2.2, has been praised by those who have so far tried it. It will support multiple indexes of a single repository (for instance, different languages), a single index of multiple repositories, or a combination.

We noted above the difficulty of changing BMech objects that were bound to other repository content. This is recognised by the Fedora community as a problem and some short time prior to the conference discussions emerged on the mailing list of how this might be tackled. At a BOF ('birds-of-a-feather') session, possibilities were discussed. It seems likely that the idea of a 'content model' will be taken out of Fedora objects and that these content models will become free-standing objects. Other Fedora objects will 'subscribe' to them. Disseminators can then be applied to the object model (with many subscribers) rather than to each subscriber object individually. This will make the development of BMechs much easier to manage. In addition, the abstraction of a content model in this way will make it possible to validate the structure of a new object for ingest against the content model definition. Objects would potentially be able to subscribe to more than one object model. This abstraction process requires the development of a definition language, but this is already underway.

There was some discussion of whether standardisation of basic datastream names between institutions might allow easy interchange of disseminators, or the establishment of a disseminator library; this is the subject of ongoing discussion. Another idea, ongoing, is that all objects should inherently support a small range of behaviours, so-called asset actions, to assist their manipulation by remote systems upon discovery.

Finally, in this section on the conference proper, we should mention Fez. Fez is a Fedora client developed in PHP aimed at the management and disclosure of completed digital objects. On the one hand it provides management tools for the author and their librarian (for want of a better characterisation), and on the other a browser interface to the repository for those treading the discovery route. It has inbuilt XACML-like security and a "powerful" user search tool. This was an impressive piece of work and we, at Hull, are considering adopting it - in the first instance for some short-term, pump-priming work. In particular we are interested in exploring Fez's new variant, 'eSpace', designed to cope with the Australian equivalent of the UK Research Assessment Exercise.

A private conversation at the end of the conference alerted us to work being done by the US National Science Digital Library (NSDL) and the University of California, Riverside. This work is developing automatic metadata creation tools around textual content and may provide a solution to RepoMMan's need for such a service.

## Meeting with University of Virginia Libraries staff

On the day immediately following the conference, the Hull delegates met with representatives from the University of Virginia (UVa) Libraries. We had requested this meeting to investigate

---

<sup>6</sup> An abbreviated version of this report will appear in the July 2006 issue of Ariadne: <http://www.ariadne.ac.uk/>

in detail the way that they used metadata in conjunction with digital objects. Our own investigations into appropriate approaches to metadata had uncovered a number of ways that the problem might be addressed, but we were aware that UVA had their own.

UVA maintains its own metadata schema covering all the data types that their digital library must cope with. This was initiated because, at the time of UVA's first digital assets, the world of metadata was very fluid and standards were still largely some way off. As time has gone on, the idea of maintaining their own schema has served them well and, now that many other standards exist, complex mappings have been developed as appropriate. In particular, all the UVA metadata can be "crumbled down to Dublin Core". This approach has its attractions for the Hull team who, likewise, envisage a repository covering a very wide range of file types. UVA have invited us to make use of their schema and to contribute to its ongoing development. This is under consideration.

## Part II: Working with Fedora web services

As noted in the introduction, the RepoMMan project was funded by JISC to develop an appropriate workflow engine to allow users to interact with a Fedora-based repository. The Project Plan identified researchers as the first group of potential users to work with and accordingly much of November and December 2005 was devoted to gaining an understanding of what this group might require of a repository<sup>7,8</sup>. The project was committed to using BPEL (Business Process Execution Language) to orchestrate Fedora web services as an underlying part of the interface.

Early work in the project<sup>9</sup> had identified an open source BPEL engine from Active Endpoints as our software of choice for this work. After some time exploring the possibilities offered by the BPEL authoring tool<sup>10</sup> and how these might be employed in an appropriate tier of services, serious work on using BPEL to orchestrate Fedora began in the spring of 2006. From the outset, the development process turned out to be less than easy.

Fedora provides two sets of web services, a SOAP-based set and a REST-based set. RepoMMan is working with the REST-based materials.

### The 'thin slice'

After the first exploratory work with BPEL, the RepoMMan team came up with the idea of developing a 'thin slice' through the three-tier stack that we eventually planned to produce for the RepoMMan tool. The diagram overleaf shows the basis of this three-tier approach.

The web-services at the top of the diagram represent just some of the services that RepoMMan's workflow tool will be required to access; Fedora's web services form part of this set, others will include services for the auto-generation of metadata and so on.

The BPEL engine orchestrates these services as necessary and interacts with the user through a model view controller (MVC) layer which we have decided to provide with Spring. The MVC layer interacts with jsp pages in the user's browser.

The 'thin slice' tested out one vertical strip through this diagram. A browser, communicating through the MVC layer with BPEL which, in turn, communicated with a single, simple service. This was essentially proof-of-concept that such an integration could be made to work and to serve our eventual purpose.

This work was completed satisfactorily in early Spring 2006.

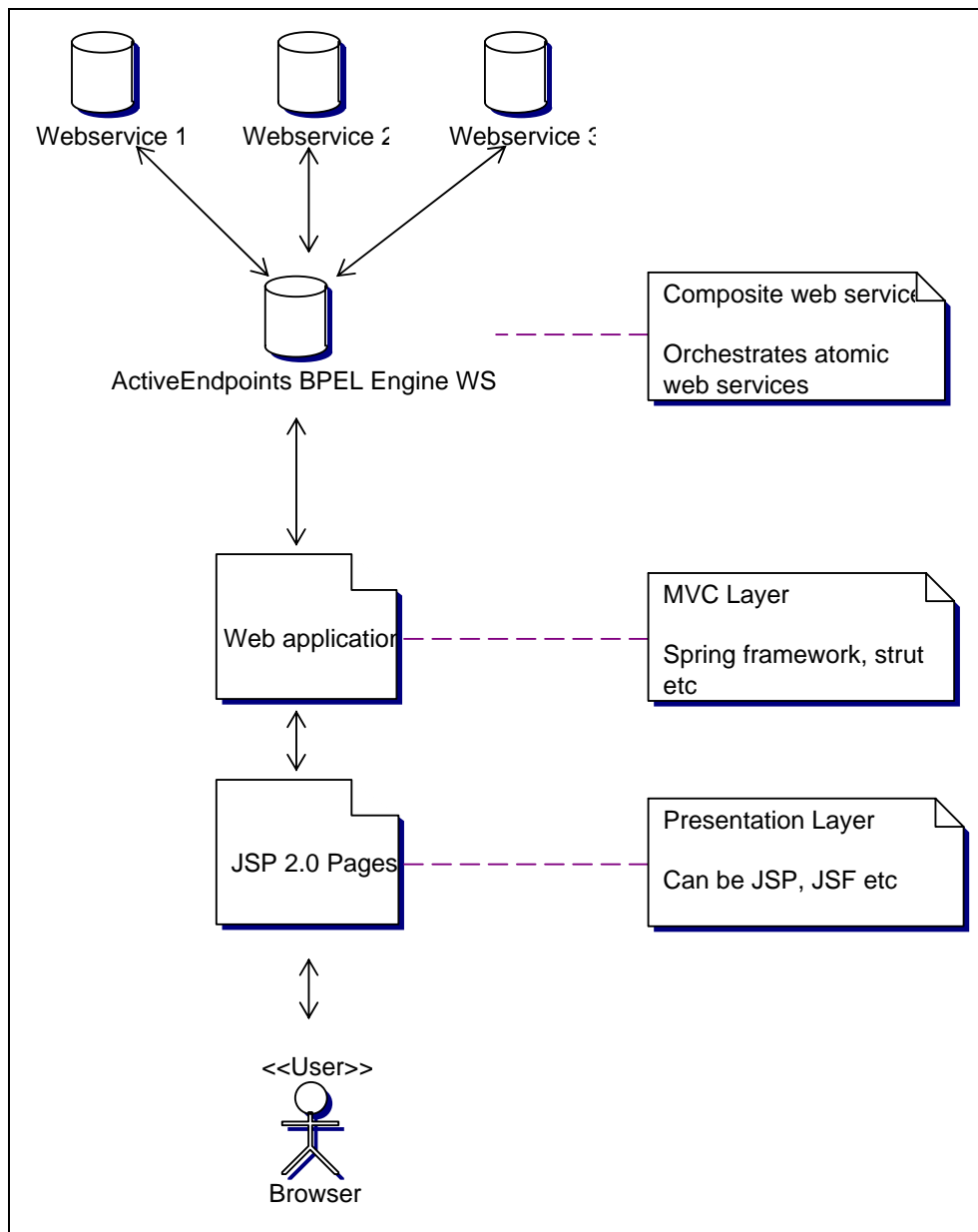
---

<sup>7</sup> RepoMMan report R-D3 Report on research user requirements survey at <http://www.hull.ac.uk/esig/repomman/documents/index.html> (valid RG 19/05/06)

<sup>8</sup> RepoMMan report R-D3 Report on research user requirements interviews at <http://www.hull.ac.uk/esig/repomman/documents/index.html> (valid RG 19/05/06)

<sup>9</sup> RepoMMan report D-D1 Available BPEL runtime environments at <http://www.hull.ac.uk/esig/repomman/documents/index.html> (valid RG 19/05/06)

<sup>10</sup> RepoMMan report D-D3 Familiarity with BPEL authoring tool at <http://www.hull.ac.uk/esig/repomman/documents/index.html> (valid RG 19/05/06)



*The RepoMMan 'three-tier stack'*

## A Fedora ingest process

Not surprisingly, our first attempt to combine BPEL and Fedora in any serious way centred around trying to ingest a simple object into Fedora 2.1. To do this, we exported an existing object from the repository in Fedora's own FOXML format and attempted to orchestrate a process to re-ingest it as a new object.

We quickly came across two problems:

The first was that Fedora's management WSDL (Web Services Description Language) file would not import into the BPEL design tool, throwing a '401 error' (unauthorised). An exchange of e-mails with the Fedora team revealed that the file itself was the subject of http security, rather than the more usual arrangement where the file was accessible to all but the service endpoints were secured. The software we are using for development does not support a process of

authentication for access to the WSDL file. This problem was overcome on a temporary basis by taking a local copy of the file so that authentication was not needed to access it.

Having made Fedora's management WSDL accessible to our software, work was started to use it. However, this proved impossible due to a second problem. The Active Endpoints BPEL software declared the WSDL file to be an illegally formed document. To check this, the file was imported into Altova's XML Spy, and into Sun's Java Studio Creator. Both agreed that the WSDL was, indeed, illegally formed.

Fedora has two WSDLs, one for management functions and the other for access. The access WSDL was tried in XML Spy with similar results. By comparison, both the WSDLs from Fedora 2.0 validated and it was possible to invoke operations using them.

E-mails to the Fedora-user list generally got some sort of response from the development team, but nothing that solved our problems.

Over the next few weeks a number of different approaches were tried to the 'WSDL problem'. We switched to what we thought would be a more straightforward process, that of retrieving information about the datastreams in an object and trying to set them all to 'inactive'. Again a range of problems was encountered with the Fedora WSDLs and another series of e-mails to the Fedora-user list provoked a response from staff at Rightscom, in London, who were starting to explore similar areas of Fedora 2.1 web services and encountering similar problems. The Fedora UK and Ireland group meeting mentioned earlier provided the opportunity to discuss matters and, as a result, over a period of days we compared notes and put together a detailed, joint e-mail to Fedora which was sent, not to the list, but to a range of key people within the project. The e-mail pointed out that the difficulties with the 2.1 WSDLs were effectively preventing either of our organisations from making any progress.

This joint e-mail was taken seriously by the Fedora team who produced a patch to fix the http security problem within a couple of days. Having looked into the issues with the contents of the WSDL files themselves they acknowledged that there was a problem. The 2.1 files use the rpc/encoded message style which a number of major software products, they accept, "have problems with". The team conceded that they should move to the document/literal style which they now see as best practice. They quickly set to work to produce a further patch.

During development, the patch was tested at Hull and at Rightscom; within the space of ten days or so a new WSDL had been produced which seemed to work as expected.

At Hull work re-started to try and develop a process that would ingest a Fedora FOXML object and this was quickly achieved. That process will now be 'grown outwards' to encompass more and more of the functionality that will be required for the RepoMMan workflow tool. Given all the time previously expended in this area, it is hoped that progress will be fairly rapid.

A concern at the back of our minds had always been that the Fedora Admin Client uses web services to communicate with the repository. Why, then, did this work in Fedora 2.1, whilst our constructions did not? This puzzle was solved at the Fedora Users' conference where it was admitted that the Fedora Admin Client 'knowing that everything works' does not attempt to validate anything.

## Part III: Conclusion

In conclusion, we are able to report that on balance we feel we have had a good experience with Fedora. Whilst the problems encountered with the 2.1 WSDL caused us great frustration at the time, the eventual outcome reflected well on the idea of a Fedora community working together to solve problems and take the project forward. With the new WSDL, work on this aspect of the RepoMMan project is now progressing steadily.

The Fedora Users' Conference coinciding, as it did, with the end of this first year was a good time to take stock of our experience and to compare it with that of others. We were pleased to come back to the UK thinking that we had made no major errors of judgement in our development; yes, there were aspects of our work where new information might modify our view but there was nothing major that we should have to backtrack on. In particular, it is likely that we shall modify our draft content models (for example, those shown here as appendices 1 and 2) in the light of discussions at the Fedora Conference and what flows from these. We shall need to take account of the move towards a set of agreed 'asset actions' and any agreement on datastream names. Likewise, the possibility of adopting the metadata schema developed by the University of Virginia may have an effect on the precise form in which we store our metadata within objects.

Fedora 2.2 will hopefully provide a few more features that we need, especially the concept of ownerId, and this will enable us to progress our thinking on the repository in general, and the workflow tool in particular, rather further. Fez offers the possibility of using Fedora for some short-term work on relatively small-scale demonstrators that may, nevertheless, have a considerable impact on the development and embedding of a repository for the University of Hull.

## References

### References

Arts and Humanities Data Service (AHDS) At: <http://www.ahds.ac.uk> (Validated Jan 2006)

Cordaro R, Daigle B, Grizzle R, Kelly J, Tuite M, Wayland R (2003) *Repository Image Object Model Committee Report* University of Virginia Library  
At: [http://www.lib.virginia.edu/digital/reports/image\\_obj\\_model\\_report.htm](http://www.lib.virginia.edu/digital/reports/image_obj_model_report.htm)  
(Validated: Jan 2006)

DCMI Usage Board (2005) *DCMI Metadata Terms* Dublin Core Metadata Initiative  
At: <http://www.dublincore.org/documents/dcmi-terms/> (Validated: Jan 2006)

DCMI Usage Board (2005) *Using Dublin Core - Dublin Core Qualifiers* Dublin Core Metadata Initiative  
At: <http://www.dublincore.org/documents/usageguide/qualifiers.shtml>  
(Validated February 2006)

Green R (2006) *D-D4 Iterative development of Fedora materials* RepoMMan Project, University of Hull  
At <http://www.hull.ac.uk/esig/repomman/documents> (valid May 2006 - RG)

Library of Congress (2004) *NISO Metadata for Images in XML Schema (MIX)*  
At: <http://www.loc.gov/standards/mix/> (Validated: Jan 2006)

Murray P (2006) *OhioLINK Image Storage* In litt. 2006-01-29

NISO (2002) *Data Dictionary - Technical Metadata for Digital Still Images* NISO

OASIS (2004) *eXtensible Access Control Markup Language (XACML) Version 2.0* Oasis Open  
At: [http://docs.oasis-open.org/xacml/access\\_control-xacml-2\\_0-core-spec-cd-04.pdf](http://docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf)  
(Validated May 2006)

Payette S, Shin E, Wilper C, Wayland R (2006) *Fedora Proposal: Content Model Dissemination Architecture* Fedora Project  
At: <http://www.cs.cornell.edu/payette/fedora/designs/cmda> (Validated May 2006)

Powell A, Day M, Cliff P (2004) *Using simple Dublin Core to describe eprints* ePrints UK  
At: <http://www.rdn.ac.uk/projects/eprints-uk/docs/simpledc-guidelines/> (Validated May 2006)

Powell A, Johnston P (2003) *Guidelines for implementing Dublin Core in XML* Dublin Core Metadata Initiative  
At: <http://www.dublincore.org/documents/dc-xml-guidelines/>  
(Validated February 2006)

Scherle R E (2006) *Image objects at Indiana State University* In litt. 06-01-29

Stanford Digital Repository (2005) *ISS-Library of Congress archive ingest and handling test (AIHT)* Stanford University Libraries and Academic Information Resources  
At: <http://www.digitalpreservation.gov/technical/aiht-stanford-final-report.pdf>  
(Validated May 2006)

Technical Advisory Service for Images (TASI)  
At: <http://www.tasi.ac.uk/> (Validated Jan 2006)

Tufts University (2005) *DCA Image Metadata* In litt. (Schavez R) 2006-01-31

Tufts University (2005) *DCA Image Specifications* In litt. (Schavez R) 2006-01-31



## Appendix 1

*Note: This is a snapshot of a document under development at a particular point in time. It is not intended to be a complete, coherent discussion, nor to be a finished document.*

# RepoMMan Project

---

INT-D3-1

University of Hull digital colour image object specification

Richard Green

January 2006

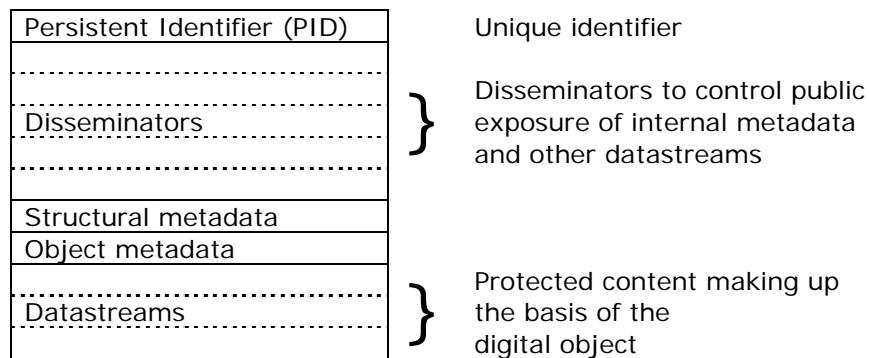
Draft 0.12 - 2006-02-12

## Object models

One of the features of Fedora that makes it attractive as the basis for a digital repository at Hull is the way that digital objects can be ascribed a range of 'disseminators' which determine how a casual user of the repository may interact with them. Defining disseminators within the repository is a non-trivial task requiring extensive programming knowledge; however, once a disseminator has been defined it can be used for any number of objects that offer a compatible internal structure. Thus, it makes sense to define a standard object model for digital images which is capable of a range of useful (to the end-user) expressions via a number of dissemination methods. The model described below has been reached after research amongst the digital repository community world-wide and it is believed that it will offer the necessary internal structure on which a full range of flexible behaviours can be built.

This is offered as a general-purpose colour image model. There will almost certainly be situations in which the needs of a particular, specialist collection of colour images require adaptation of this format. The specification may not satisfy the University's needs in terms of bitonal images for which another model may need to be developed.

A standard Fedora object model may be represented in the following way:



Let us consider these elements working from the 'bottom' up.

### Content datastreams

The standard UoH colour image model will consist of content datastreams with the following IDs and content as defined below.

#### fullSize

A mandatory datastream containing a full-size version of the original in jpeg format. Users requiring other formats can convert the image themselves using appropriate software or, perhaps, by invoking a disseminator which can assume a jpeg original.

#### screenSize

A mandatory datastream providing a jpeg image to fit within a box 800px wide by 600px deep. Most browsers will further scale it, if necessary, to fit the user's screen dimensions. If the original image (FullSize) is smaller than this recommended dimension then 'FullSize' and 'ScreenSize' will both take the dimensions of the original.

#### thumbnail



Format	MIME type
Identifier	
Source	
Language	
Relation	
Coverage	
Rights	(qualified by 'access rights' and 'licence')

## MIX:

## Format details including MIMEType

FileSize	(This seems not to be included in the JHOVE MIX output but is provided elsewhere in the file)
ScannerManufacturer	(This may be 'unknown' but a value would indicate a scan)
DigitalCameraManufacturer	(This may be 'unknown' but a value would indicate a photo)
CameraCaptureSettings	(To be empty, but a placeholder for EXIF information in the future - JHOVE provides all the data but does not itself write it to the MIX metadata block)

## DateTimeCreated

## Image structural details including

Resolution	
ImageWidth	(The image width of 'FullSize' in pixels)
ImageLength	(The image length/height of 'FullSize' in pixels)

Thus a record for a photographic image might be:

```
<UoHMetadata>
<oai_dc:dc xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dcq="http://purl.org/dc/terms/">
  <dc:title>The Roman aqueduct near Tarragona, Spain</dc:title>
  <dc:creator>Richard Green</dc:creator>
  <dc:subject>Architecture, Roman</dc:subject>
  <dc:description>The Roman aqueduct near Tarragona, Spain. The 'Pont du
Diable' (The Devil's Bridge) is 27m/ca. 90ft tall at its highest point and
217m/ca. 600ft long. The conduit at the top was originally covered but now you
can walk along it.</dc:description>
  <dc:publisher>University of Hull</dc:publisher>
  <dc:contributor></dc:contributor>
  <dc:date>2006-01-27</dc:date>
  <dcq:available>2006-01-27 - </dcq:available>
  <dc:type>image</dc:type>
  <dc:format>image/jpeg</dc:format>
  <dc:identifier>Hull:1</dc:identifier>
  <dc:source>University of Hull Photographic Service</dc:source>
  <dc:language></dc:language>
  <dc:relation></dc:relation>
  <dc:coverage>Barcelona, Spain</dc:coverage>
  <dc:rights>©2005 Richard Green</dc:rights>
  <dcq:accessRights>UoH</dcq:accessRights>
  <dcq:licence>none</dcq:licence>
</oai_dc:dc>
<mix:mix xmlns:mix="http://www.loc.gov/mix/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.loc.gov/mix/ http://www.loc.gov/mix/mix.xsd">
<mix:BasicImageParameters>
  <mix:Format>
    <mix:MIMEType>image/jpeg</mix:MIMEType>
    <mix:ByteOrder>little-endian</mix:ByteOrder>
    <mix:Compression>
```

```

    <mix:CompressionScheme>1</mix:CompressionScheme>
  </mix:Compression>
  <mix:PhotometricInterpretation>
    <mix:ColorSpace>2</mix:ColorSpace>
    <mix:ReferenceBlackWhite>0.0 255.0 0.0 255.0 0.0
      255.0</mix:ReferenceBlackWhite>
  </mix:PhotometricInterpretation>
  <mix:Segments>
    <mix:StripOffsets>350</mix:StripOffsets>
    <mix:RowsPerStrip>4294967295</mix:RowsPerStrip>
    <mix:StripByteCounts>921600</mix:StripByteCounts>
  </mix:Segments>
  <mix:PlanarConfiguration>1</mix:PlanarConfiguration>
</mix:Format>
<mix:File>
  <mix:FileSize>922262</mix:FileSize>
  <mix:Orientation>1</mix:Orientation>
</mix:File>
</mix:BasicImageParameters>
<mix:ImageCreation>
  <mix:ScanningSystemCapture>
    <mix:ScanningSystemHardware>
      <mix:ScannerManufacturer>SONY</mix:ScannerManufacturer>
      <mix:ScannerModel>
        <mix:ScannerModelName>CYBERSHOT</mix:ScannerModelName>
      </mix:ScannerModel>
    </mix:ScanningSystemHardware>
  </mix:ScanningSystemCapture>
  <mix:DigitalCameraCapture>
  </mix:DigitalCameraCapture>
  <mix:CameraCaptureSettings>
  </mix:CameraCaptureSettings>
  <mix:DateTimeCreated>2003-05-16T16:31:27</mix:DateTimeCreated>
</mix:ImageCreation>
<mix:ImagingPerformanceAssessment>
  <mix:SpatialMetrics>
    <mix:SamplingFrequencyUnit>2</mix:SamplingFrequencyUnit>
    <mix:XSamplingFrequency>72</mix:XSamplingFrequency>
    <mix:YSamplingFrequency>72</mix:YSamplingFrequency>
    <mix:ImageWidth>640</mix:ImageWidth>
    <mix:ImageLength>480</mix:ImageLength>
  </mix:SpatialMetrics>
  <mix:Energetics>
    <mix:BitsPerSample>8,8,8</mix:BitsPerSample>
    <mix:SamplesPerPixel>3</mix:SamplesPerPixel>
  </mix:Energetics>
</mix:ImagingPerformanceAssessment>
</mix:mix>
</UoHMetadata>

```

[Note 1: the MIX metadata in the example was not obtained from the image that provided the DC data and is for illustrative purposes only]

[Note 2: although this metadata relates to a TIFF image from a digital camera, JHOVE is reporting it as scanned. Are we misunderstanding the apparent MIX distinction between scanner and camera or is JHOVE wrong?]

**Can the MIX meta hold the data about the archive version as 'original'? Do we need a different MD stream?**

**Can we write a routine to 'dumb down' the MIX data to just our needs?**

## Disseminators

Before disseminators can be constructed, it is necessary to define the set of behaviours that will be expected of them. At this stage we might suppose two disseminators: one related to the digital content and one related to the metadata.

### Content dissemination

At the very least, the following self-explanatory behaviours will be required:

- getThumbnail
- getScreenSize
- getFullSize

and one might postulate a range of further dissemination methods

- brightImage
- convertImage (to gif/tif/png/bmp)
- cropImage
- grayscaleImage
- resizeImage
- watermarkImage
- zoomImage

These behaviours are amongst those provided by Fedora 'out-of-the-box' and could easily be attached to the FullSize or ScreenSize image - but not both. That said, they appear to depend on a remote connection to the UVa and so perhaps we ought to ask about legality and sustainability first!

It may seem strange to define a set of behaviours (getThumbnail etc) that simply retrieve datastreams directly. However, by doing this we remove the need for 'guest' users of the repository to access *any* datastreams directly. It can be arranged that they can *only* retrieve data through a disseminator - thus considerably restricting their access to digital objects.

### Metadata dissemination

It seems likely that the following list will be modified as development proceeds, but to begin with: **[Can this disseminator be generic - for non-images too?]**

- getPublicMetadata
  - retrieve all the image metadata that could be useful to a user - ie the entire metadata set above: the contents of UoHMetadata
- getDescription
  - get the basic descriptive metadata (only)
    - title, creator, subject, description, publisher, date
- getSource
  - get the source and provenance metadata (only)
- getTechnical
  - get the technical metadata (only), ie the MIX data
- getRights
  - get the metadata relating to copyright, IPR etc (only)
    - rights, access rights and licence

The last four disseminators retrieve subsets of the metadata retrieved by the first.

## Security

At this stage in the development of our repository work it seems that a number of our (eventual) security measures may relate to the 'content model' of an object. Fedora builds the concept of a content model into the immutable bit of an object's metadata when it is created. This being the case, it will be necessary to have a standard name for the content model of the University's public-facing images of this type. It is suggested that this be **UoH\_Std\_Col\_Img**.

## Filenames

Where possible, filenames should conform to a standard format. This has two purposes: firstly that from its name it should be straightforward to identify what function the file serves within the repository and, secondly, that any file that becomes displaced from its home directory can be located and reinstated.

The following is proposed:

UoHPS-060211-0001-sc-optionalname.jpg

- The first block identifies the home directory for the file and relates to the collection or owner: UoHPS is the University Photographic Service
- The second block is the creation date: yymmdd.
- The third block is a serial number, allowing up to 9999 files to be created within a particular space each day.
- The fourth block identifies function: 'sc' is a screen-size image (cf 'ar', 'fs' and 'th').
- The final block is an optional filename or descriptor.



## Appendix 2

*Note: This is a snapshot of a document under development at a particular point in time. It is not intended to be a complete, coherent discussion, nor to be a finished document.*

# RepoMMan Project

---

INT-D3-3

University of Hull public document object specification

Richard Green

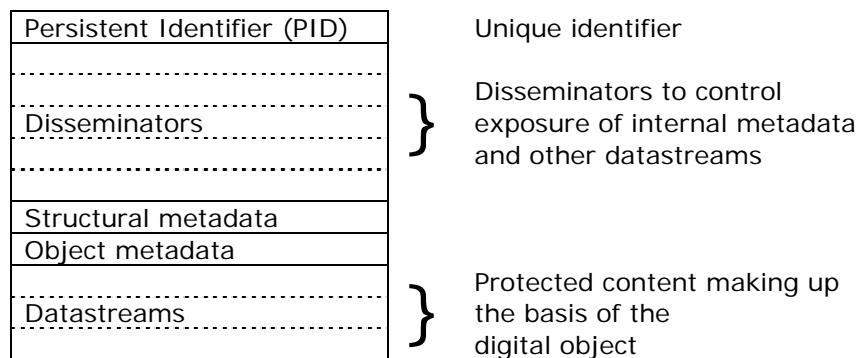
February 2006  
Draft 0.1

## Object models

One of the features of Fedora that makes it attractive as the basis for a digital repository at Hull is the way that digital objects can be ascribed a range of 'disseminators' which determine how a casual user of the repository may interact with them. Defining disseminators within the repository is a non-trivial task requiring extensive programming knowledge; however, once a disseminator has been defined it can be used for any number of objects that offer a compatible internal structure. Thus, it makes sense to define a standard object model for digital images which is capable of a range of useful (to the end-user) expressions via a number of dissemination methods. The model described below has been reached after research amongst the digital repository community world-wide and it is believed that it will offer the necessary internal structure on which a full range of flexible behaviours can be built.

This document offers a general-purpose public document model. That is to say a model for a document that is intended for exposure to the public and to harvesting processes.

A standard Fedora object model may be represented in the following way:



Let us consider these elements working from the 'bottom' up.

### Content datastreams

The standard UoH public document model will consist of content datastreams with the following IDs and content as defined below.

#### **document.xxx**

This datastream will hold the 'original' version of the document and will not be made available to a normal user of the repository. 'xxx' is the file extension of the format used. Thus a document that originates in a Microsoft Word format becomes 'document.doc'. This is, in effect, an archive copy.

#### **document.pdf**

Documents will generally be made available to users in pdf format, and the 'document.pdf' datastream will contain the version of the document normally made available.

### Other datastreams

There may be other datastreams that relate to the functionality of the disseminators associated with the object. These are internal constructs that would not be exposed to a user.



## Disseminators

Before disseminators can be constructed, it is necessary to define the set of behaviours that will be expected of them. At this stage we might suppose two disseminators: one related to the digital content and one related to the metadata.

### Content dissemination

At the very least, the following self-explanatory behaviours will be required:

- getDoc (not available to the public)
- getPdf

It may seem strange to define a set of behaviours (getPdf etc) that simply retrieve datastreams directly. However, by doing this we remove the need for users of the repository, be they University members, or 'guest' users, to access *any* datastreams directly. It can be arranged that they can *only* retrieve data through a disseminator - thus considerably restricting their access to digital objects and ensuring that the primary data is protected from arbitrary or accidental alteration.

### Metadata dissemination

Fedora 2.1 is capable of holding multiple metadata formats in the same datastream. Thus 'UoHMetadata' can hold both DC metadata for general use and for OAI harvesting, and an RDF extended metadata section. The different types can be exposed as needed using an appropriate disseminator. In addition the same datastream can hold provenance information to go with each metadata type. We might envisage a range of disseminator methods:

- getPublicMetadata
  - retrieve all the object metadata that could be useful to a user - ie the entire metadata set above: the contents of UoHMetadata
- getMetadata (argument)
  - retrieve metadata of a particular type from UoHMetadata
- getDescription
  - get the basic descriptive metadata (only)
    - title, creator, subject, description, publisher, date
- getProvenance
  - get the source and provenance metadata (only)
- getTechnical
  - get the technical metadata (only)
- getRights
  - get the metadata relating to copyright, IPR etc (only)
    - rights, access rights and licence

The last five disseminators retrieve subsets of the metadata retrieved by the first. Note that these methods would be appropriate to many types of digital object (not just document objects) and a suitable constructed metadata disseminator could potentially have broad application within the repository.

## Security

At this stage in the development of our repository work it seems that a number of our (eventual) security measures may relate to the 'content model' of an object. Fedora builds the concept of a content model into the immutable bit of an object's metadata when it is created. This being the case, it will be necessary to have a standard name for the content model of the

University's public-facing objects of this type. It is suggested that this be **UoH\_Public\_Document**.

## Filenames

Where possible, filenames should conform to a standard format. This has two purposes: firstly that from its name it should be straightforward to identify what function the file serves within the repository and, secondly, that any file that becomes displaced from its home directory can be located and reinstated.

The following is proposed:

repomman-060211-0001-optionalname.pdf

- The first block identifies the home directory for the file and relates to the collection or owner.
- The second block is the creation date: yymmdd.
- The third block is a serial number, allowing up to 9999 files to be created within a particular space each day.
- The final block is an optional filename or descriptor.

## Appendix 3

### Fedora UK and Ireland User Group membership - June 2006

Chris Awre	RepoMMan Project	University of Hull
Jack Bazuzi		VTLS
Paul Bevan		National Library of Wales
Ian Dolphin	RepoMMan Project	University of Hull
Martin Dow		Rightscom Ltd
Renhart Gittens	Paradigm Project	University of Oxford
Richard Green	RepoMMan Project	University of Hull -- Group coordinator
Eric Jutrzenka	IRIScotland	National Library of Scotland
Gareth Knight		Arts & Humanities Data Service
Simon Lamb	RepoMMan Project	University of Hull
Boon Low		UK National e-Science Centre, Edinburgh
John McDonough	Irish Virtual Research Library & Archive	
		University College, Dublin
Dave Price		University of Oxford
Glen Robson		National Library of Wales
Sally Rumsey		London School of Economics
Ben Ryan		Kainao Ltd
Robert Sherratt	RepoMMan Project	University of Hull
Jackie Spence		University of Wales, Aberystwyth
Susan Thomas	Paradigm Project	University of Oxford
Dave Thompson	Wellcome Library	Wellcome Trust
Iain Wallace	Spoken Word Services	Glasgow Caledonian University

