

**Scoping Study on
Repository Version Identification
(RIVER)
Final Report**

Submitted by:	Sally Rumsey & Frances Shipsey LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE Michael Fraser & Howard Noble OXFORD UNIVERSITY COMPUTING SERVICE Mark Bide, Hugh Look & Deborah Kahn, RIGHTSCOM
Submitted to:	Fred Friend JOINT INFORMATION SYSTEMS COMMITTEE Working Group on Scholarly Communications
Version:	V.05 Draft Final Report
Date:	31 March 2006

Rightscom Ltd
Lincoln House
75 Westminster Bridge Road
London SE1 7HS
UNITED KINGDOM
Tel: +44 20 7620 4433
www.rightscom.com

Contents

Contents	2
1 Introduction – scoping the version identity issue	3
1.1 Developing an initial vocabulary	3
1.2 The requirement to identify versions	6
1.3 Version numbering and version granularity	6
1.4 DigitalEditions and lifecycle semantics	7
1.5 DigitalEditions and access/usage privileges	9
1.6 Version identification and complex, dynamic objects	9
1.7 Authenticity and trust	10
2 Some alternative perspectives on version identity	11
2.1 Authors and other creators	11
2.2 Publishers	11
2.3 Users	12
3 Scenarios for RIVER report	13
3.1 Use Case collections	13
3.2 Scenarios developed by the RIVER Project	13
3.3 Use cases submitted to the RIVER workshop on 15 Feb 2006	19
3.4 Summary of requirements from scenarios and use cases	20
4 Review of current repository practice	23
4.1 Introduction	23
4.2 Methodology	23
4.3 Repositories and content types	24
4.4 Case studies	25
4.5 Identity and version management in the OAI Protocol (short note)	34
5 Mechanisms we have found in use for fulfilling the identified requirements	36
5.1 Introduction	36
5.2 Disambiguation and collocation	36
5.3 Access and authorisation control	36
5.4 DigitalEdition identification	37
5.5 Workflow	37
6 Conclusions	38
7 Recommendations	40
7.1 JISC	40
7.2 Universities (Institutional Repository managers)	41
Appendix 1: Sources consulted	42
JISC-funded Projects	42
References	42
Appendix 2: Participants at workshop held 15 Feb 2006	44

1 Introduction – scoping the version identity issue

1.1 Developing an initial vocabulary

At first sight, the question of identification of different versions of digital objects held in institutional repositories may appear to be a relatively simple one. However, it is not long before every investigation of identity comes up against a common set of very complex issues which – tempting though it may be – cannot simply be set to one side. While this report is essentially about identifying current practice and recommending improvements, it must start from some consideration of the theoretical framework within which such improvements can be robustly formulated.

Nevertheless, this is not the place for an extended consideration of the history and philosophy of identity. Rather, we will try to illustrate some fundamental issues through the use of examples.

The first of these is a concept that has come to be known¹ as “functional granularity”. The basis of this principle is that decisions about whether to identify two different objects as being “the same thing” (that is, as both being members of a common class) or as being “different things” (members of different classes or different instances of the same class) are only possible in the light of functional requirements.

There is a very simple example of this from within our every day experience. For the most part, it is entirely satisfactory to identify an entire class of book with an ISBN. Two copies of “the same book” are for the most part functionally identical. But there are circumstances where this is not true, perhaps in a library, where the identification of an individual instance of “a book” may be an essential element of good management.

This issue of functional granularity is one which we cannot avoid coming up against repeatedly, particularly in considering any concept of identifying a “version” of something since, by its very nature, a “version” will be from some perspectives in the same class as the object from which it is derived, and from some other perspectives in a different class. The question is therefore always one of perspective driven by functional requirement. We explore the question of viewpoint in more detail in Section 2 of this report.

This leads us to a second principle, which is that questions of identity lie ultimately in the social and policy sphere, rather than in the technical one. If identification is going to be a useful tool for communication, then we must first agree on the key identity issues. While unique and persistent identification of digital objects is critical for effective machine-to-machine communication, technology can only help (or, at worst, hinder) the application of identification policies – we cannot look to technology to tell us what the policy should be (beyond the trivial level). So, we may decide that a workflow, time-based model of versioning works well within a particular framework: in these circumstances, time-stamping and version number incrementing systems clearly have a value – but they tell a very limited story.

This leads us to consideration of another principle, which is that to talk of something as a “version” implies that it has a relationship with the thing of which it is a “version”. We need to have some language with which to speak about what that relationship might be, and have found some useful generic terminology

¹ After the <indecs> project, www.indecs.org

in the WebDAV specification which we have adopted for use in this document.² This set of semantics (as is evident from the definitions) relates to a workflow-controlled versioning system; as a result, it can provide us with some useful terminology for talking about time-based approaches to version identity.

Table 1. Some useful version semantics from the WebDAV specification

WebDAV Term	WebDAV definition [Edited]
Predecessor Successor Ancestor Descendant	<p>When a version-controlled resource is checked out and then subsequently checked in, the version that was checked out becomes a "predecessor" of the version created by the check in. A client can specify multiple predecessors for a new version if the new version is logically a <i>merge</i> of those predecessors.</p> <p>When a version is connected to another version by traversing one or more predecessor relations, it is called an "ancestor" of that version. The reciprocal of the predecessor and ancestor relations are the "successor" and "descendant" relations. Therefore, if X is a predecessor of Y, then Y is a successor of X, and if X is an ancestor of Y, then Y is a descendant of X.</p>
Root Version	The Root Version is the version in a version history that is an ancestor of every other version in that version history.
Fork Merge	When a second successor is added to a version, this creates a "fork" in the version history. When a version is created with multiple predecessors, this creates a "merge" in the version history. A server may restrict the version history to be linear (with no forks or merges), but an interoperable versioning client should be prepared to deal with both forks and merges in the version history.

However, consideration of the issue of versioning purely from a workflow perspective risks being seriously misleading. There is an inbuilt tendency to assume that a "later" version is also a "better" version. However, the question of whether a Successor is better is always a matter of point of view.

From a purist document workflow viewpoint, a document may be evolving and each Successor may indeed be an enhancement of its Predecessor. However, some types of Successor may actually not be enhancements – as, for example, when a thumbnail is created from an image file: from a functional viewpoint this may be "better" (as in more fit for a specific purpose) but, from another point of view, it is clearly an inferior image.

This brings us to another issue, which was underlined for us by consideration of a number of Use Cases, particularly those which have been usefully published by Johns Hopkins University (gathered for their ProjectRepository).³

In looking at these Use Cases, we recognised and have tentatively named four distinctive uses of the term "version" and, although these do not have entirely impermeable edges, they again provide us with a set of semantics which will be useful to our discussion. In order to make sense of this, though, we have to introduce a particularly slippery concept into the discourse, the idea of the

² See *Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)* <http://www.webdav.org/specs/rfc3253.html>; this should not be read as an endorsement of the specification, nor a view that the terminology adopted by WebDAV should be regarded as definitive.

³ Currently to be found at <https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository> but note that this is a work in progress, and this version may not be stable...

“information payload” of a digital object. This is probably an easier idea to grapple with than the idea of a “work”.⁴

Table 2. A tentative typology of “versions”, as defined by the RIVER project team

RIVER Term	RIVER definitions	Comments and examples
DigitalCopy	A Successor digital object that is identical to its Predecessor digital object in all significant attributes, and in particular with respect to its information payload and technical file format	What we would commonly just call a copy. For example, I am writing in a Word file. If I were to save this file locally, and then make a copy on a remote server, that version on the server would be a DigitalCopy
DigitalVariant	A Successor digital object that is essentially identical to its Predecessor digital object with respect to its information payload but is in a different technical file format	This is perhaps most easily illustrated in terms of graphic file formats – for example a JPEG file derived from a TIFF file of a photograph. Such transformations can, of course, be lossy from the point of view of the quality of the information payload that they carry, but they continue to be (at some level) the same photograph. Another example might be if I convert this Word file (without changing its content) into a PDF.
DigitalRevision	A Successor digital object whose information payload has minor changes but where its reviser decides that there is no need for the changes to be recognised by renaming.	For example, if I make some small editorial corrections to a Word document and save it without giving it a new name.
DigitalEdition	A Successor digital object which has information payload which has evolved from the content of its Predecessor digital object	This is the time-based “workflow” relationship most commonly associated with the term “version”; examples are a preprint and a postprint of the same journal article.
DigitalEquivalent	A relationship between two digital objects where this is no Ancestor/Descendant relationship, but where nevertheless one may be functionally equivalent to the other in a given context	This is a relationship which is likely to be increasingly significant in relation to complex courseware. There is a valuable Use Case in the Johns Hopkins collection relating to the requirement for use of learning objects which are culturally appropriate to the individual User. Two learning objects might therefore have absolutely no common information payload but would fulfil exactly the same purpose.

⁴ Work is a term that is commonly used for much the same idea – see, for example *Functional Requirements for Bibliographic Records (FRBR)* <http://www.ifla.org/VII/s13/frbr/frbr.htm>. However, the use of the term “work” is particularly problematic, not least because of its relationship to copyright law (where its definition is different from the FRBR definition). Since we would necessarily have to create a third local definition, we prefer to use the term “information payload”.

We explicitly see all these different types of version as being within the scope of any consideration of the identification of versions in institutional repositories.⁵

1.2 The requirement to identify versions

We ultimately can recognise only two underlying reasons why it should be necessary for a user (defined in the broadest sense to include both people and systems) to need to identify versions. We have defined these in Table 3. This terminology proves useful in the analysis of Use Cases, since it can help to demonstrate that two apparently dissimilar Use Cases are in reality expressions of the same requirement.

Table 3. The two broad classes of requirement for version identification, as defined by the RIVER project team

RIVER Term	RIVER definitions	Comments and examples
Collocation	The act of identifying that two digital objects have a contextually meaningful relationship [without inspecting and comparing the objects themselves]	To enable a user to identify digital objects that may be functionally equivalent in a given context. For example, to know that two objects in two different open access repositories are in fact DigitalCopies of a common ancestor.
Disambiguation	<ul style="list-style-type: none"> The act of identifying that two digital objects which happen to share certain attributes (eg the same title) have no contextually meaningful relationship 	To enable a user to discriminate between different digital objects which are apparently similar (have similar attributes: for example the same title and author) but are not necessarily functionally equivalent in a given context. This is a generic version of the “appropriate copy” question – what is “the best” version for any specific use/user: for example, because it’s a copy of the published version; or because access to it is free.
	<ul style="list-style-type: none"> The act of understanding the meaning of the relationship between two digital objects where one exists [without inspecting and comparing the objects themselves] 	

1.3 Version numbering and version granularity

The extent to which version granularity is a policy issue is perhaps best illustrated by consideration of the identification of software versions.

When an amendment is made to version 1.8.0, is the next version 1.8.1, 1.9.0 or 2.0.0? That decision is very clearly a policy decision (in essence, often a

⁵ For what we believe will be obvious reasons, we have decided to remain outside the debate on the definition of an “Institutional Repository”. To the extent that we draw boundaries at all, we draw them deliberately broadly. Our approach appears to be in line with one JISC definition of a repository as “a place where a range of digital materials may be stored, and which does have an associated rationale in terms of what is being stored and what usage the repository is set up to serve”. See Carpenter (2005) *Repositories in Context* <http://www.ukoln.ac.uk/events/delos-rep-workshop/presentations/carpenter.ppt>

marketing decision); it may be entirely arbitrary, or it may be based on an objective criterion (for example, that v2.0.0 marks an end to backwards compatibility with v1).

1.4 Digital Editions and lifecycle semantics

There are two projects currently in hand of which we are aware that are considering mechanisms for identifying versions. The two are both working in the same general area – journal articles. The JISC VERSIONS project (*Versions of Eprints – a user Requirements Study and Investigation Of the Need for Standards*) is led by the London School of Economics and Political Science and “addresses the issues and uncertainties relating to versions of academic papers in digital repositories” with specific reference to papers in economics. It is focussing particularly on issues relating to trust, but is still at an early stage of its work (it is due to deliver its final report in January 2007). It is therefore a little early to draw any conclusions from its work (although we have had valuable access to some draft Use Cases).

Of more immediate significance, perhaps, is the joint NISO/ALPSP *Working Group on Versions of Journal Articles* which has as its substantive deliverables:

- A preferred vocabulary for the most common life cycle stages [of journal articles]
- Development of appropriate metadata to identify each variant version⁶ and its relationship to other versions, in particular the definitive, fully functional published version
- Establishment of practical systems for ensuring that the metadata is applied by authors or repository managers and publishers

Although the Working Group has yet formally to report, we have been given access to the draft semantics which have been circulated for review. While these do not represent the definitive output from the project, they are particularly useful as an example of an approach to definitions of stages of a workflow (see Table 4).

Table 4. Draft NISO/ALPSP journal article lifecycle semantics

NISO/ALPSP Term	NISO/ALPSP definition
Author's original	A version of a journal article that is considered by the author to be of sufficient quality to be submitted for review by a second party. This review may be prior to any formal review for publication. The author accepts full responsibility for the article. May have a version number or datestamp. Content and layout as set out by the author.
Accepted manuscript	The version of a journal article that has been accepted for publication in a journal. A second party (the “publisher” – see “Version of Record” below for definition) takes responsibility for the article. Content and layout as submitted by the author.

⁶ Note: their semantics, not ours!

NISO/ALPSP Term	NISO/ALPSP definition
Proof	A version of a journal article that is created as part of the publication process. This includes the copy-edited manuscript, galley proofs (i.e. a typeset version that has not been made up into pages), page proofs, and revised proofs. Some of these versions may remain essentially internal process versions, but others are commonly released from the internal environment (e.g. proofs are sent to authors) and may thus become public, even though they are not authorised to be so. Content has been changed from Accepted Manuscript; layout is the publisher's.
Version of record	A version of a journal article that has been made available by any organization that acts as a publisher by declaring the article "fit for publication" (i.e. the publisher). This includes any "early release" articles that are formally identified as being published.
Updated version of record	A version of the Version of Record of a journal article that has been amended in some way.

There are several interesting features of this set of semantics. The first lies in the evident difficulty that this eminent group has had in developing clear and unambiguous definitions of the stages in the lifecycle. The second is the difficulty in finding appropriate semantics, particularly in what has proved to be a fairly difficult debate. The choice of the almost legalistic (and definitely semantically significant) term "version of record" in what is otherwise a lifecycle sequence is a clear case in point.

A similar approach to defining specific stages in the lifecycle as proposed by the NISO/ALPSP working group appears to be favoured by the Open Access community:

- **"Preprint" : "Postprint"**
- **"Peer reviewed" : "Not peer reviewed"**
- **"Submitted for publication" : "Unpublished" : "Published"**

So far as we can tell, these attributes are proposed without formal definition. While this avoids the difficulty of creating definitions, it depends on the view that "everyone knows what x means" – and as an approach this tends to break down fairly quickly, creating ambiguity. That potential ambiguity is even more obvious in tags such as "minor amendments made".

Despite the obvious challenge, lifecycle semantics have considerable value for disambiguation in cases where agreement can be reached on common lifecycle terminology and definition. It is not only journal articles that might be addressed in this way – another obvious Use Case is the e-thesis.

However, this fails to answer a key question: how can a "version history" be established, through which the different DigitalEditions which are Descendants of a common Root Version can be linked to one another. This depends either on robust identification of the Root Version being carried through the entire workflow – something which may be easier in the case of an e-thesis than it is in the case of an e-journal article – or on matching some other attributes (title, author) which can only be uncertain (what proportion of journal articles change their title during the publication process?).

Human mediation of the collocation of Ancestors with their Descendants offers a partial solution, but is not really scalable. Search engines may give “good enough” results for some purposes but perhaps not for others.⁷

So, while standard lifecycle semantics may have a significant contribution to make, they can only ever address part of the problem.

1.5 DigitalEditions and access/usage privileges

The owner of a digital resource may wish to distinguish between different classes of user and the permissions that these may have at different points in the lifecycle of the resource. This provides yet another dimension to the version identification semantic challenge, in terms of the specific permissions (eg See, View, Read, Write, Delete) which may be granted to different classes of user.

In some circumstances, different DigitalEditions might (from the point of view of a potential user) be most usefully distinguished by terms that define their permission status:

- **Published** - revision and version control are beyond the control of the author and changes can only be made in special circumstances (for example, if the resource contains defamatory material)
- **Open** - a resource is openly available but can branch and be updated at any time
- **Protected** - the access to a resource is controlled. Only defined groups and individuals will be able to gain access to the resource and for each it will be predefined as to what permissions they have with respect to the resource
- **Closed** - only the Author(s)/Owner(s) can gain access to a resource

This is not intended to be a definitive list of such states, nor as a proposed semantic structure; however, it serves to illustrate the way in which authentication and authorisation considerations interact with the mechanisms used to identify a resource and versions of a resource, and how this is subject to change over time.

1.6 Version identification and complex, dynamic objects

There are particular (and very well known) problems with the identification of dynamic resources which cannot be ignored. While it is possible to identify “a database” or “a Wiki”, any concept of a “version” of either – except perhaps by reference to a particular time – is perhaps of doubtful value.⁸

However, we have seen a particularly challenging set of Use Cases relating to complex learning objects, particularly where these are simply collections of references “out” to remote resources (which may themselves be dynamic).⁹ What is the functional granularity of identity in these circumstances – at what point do you re-identify a Learning Object as being a Successor?

⁷ For a useful discussion of the complexities of collocation and disambiguation when it comes to automating citation analysis, see Peter Jasco’s article, “Google Scholar and the Scientist”: <http://www2.hawaii.edu/~jacso/extra/gs>

⁸ This raises the general question of “persistence”, which has come up in some Use Cases we have considered. We have made a very clear distinction between the persistence of the identifier (in other words, that an identifier should always identify the same referent) and the persistence of the referent. There are clearly some critical policy issues for Repositories on whether they permit deletions, but these are out of scope for this report.

⁹ We have already mentioned the issue of the DigitalEquivalent, where completely different Learning Objects may fulfil the same function.

The questions that come into play are very different from those relating to a workflow view of version identity. To the extent that they fulfil identical learning objectives, there are strong arguments that two learning objects with completely different content might most appropriately (from the point of view of the user, at least) be identified as being “the same thing”.

1.7 Authenticity and trust

No scoping of identity issues can be complete without some mention of authenticity and trust.

Ultimately, the user needs to understand (or at the very least to be able to deduce) *who says*, for example, that this “postprint” is an accurate representation of the “published version”. Unless users can trust that the information that they are being given about version identity is accurate, it is valueless.

2 Some alternative perspectives on version identity

We have not undertaken any sort of formal requirements gathering exercise as part of the RIVER project. We are therefore to some extent speculating on what the likely requirements of different stakeholders are likely to be.

2.1 Authors and other creators

It is certainly arguable that it is only authors¹⁰ who can authoritatively make statements about different editions, certainly in terms of which DigitalEdition most accurately represents their current view of which is “best”. It is also authors who are most likely to be able to make definitive statements of which versions are related to one another and what that relationship actually is.

Authors’ primary interests in version identification are likely to be:

- To ensure collocation of all the versions of their creation that they wish to disseminate; this is likely to be particularly significant in managing processes such as citation analysis and other measures, which are particularly important to most authors
- To ensure that their moral right of paternity is properly exercised – in almost all circumstances, authors expect to be recognised as such

Since they are at the beginning of the creation process, it is self evident that, ideally, authors should uniquely identify their own works, and that that identity, once established, should remain attached to the work throughout its life. Indeed, this was recognised as “the ideal” in the development of the International Standard Text Code¹¹ (ISTC) which has been specified for the identification of textual works.

However, the ISTC has still to become an international standard, because of difficulties with the establishment of a commercial model for its implementation and the related issue of appointing a Registration Authority. Even if the ISTC does finally become an international standard – and that is likely to become clear during 2006 – considerable difficulties can be anticipated in persuading authors to register works, and then in ensuring that this identity is maintained through the lifecycle.

2.2 Publishers

Anyone who provides content that is publicly accessible is “a publisher” by any properly coherent definition of that term; therefore, anyone who runs an institutional repository is a publisher. However, to use the term in this generic sense would be confusing, so by “publisher” here we mean the type of organisation (both commercial and non-commercial) that have conventionally been called “publishers”.

These have been the organisations that have traditionally been responsible for assigning unique identification to publications, although their interest in identification rarely extends beyond the assignment of “supply chain” identifiers, of which the classic exemplar is the ISBN. Almost all books worldwide are assigned ISBNs, and it is probably the most effective identifier of its type.¹²

¹⁰ Wherever we use this term, we include all types of creators.

¹¹ See <http://www.collectionscanada.ca/iso/tc46sc9/wg3.htm>

¹² Bearing in mind that over 200,000 new ISBNs were assigned in the UK during 2005.

The ISSN, which is used to identify journals, is in a slightly different category, since it is essentially a bibliographic identifier. However, combined with dates it is also often used as a supply chain identifier, and the majority of serials which are published for the academic market (journals) have ISSNs assigned.

A significant proportion of published journal articles now have Digital Object Identifiers (DOIs) assigned at the article level by their publisher; no other article-level identifier has achieved a similar level of market adoption (although several others have been in more-or-less widespread use). The DOIs assigned by CrossRef (the DOI registration agency which focuses entirely on the identification of journals and journal articles) identify journal articles at the “work” level (as distinct from any specific manifestation of the article, either in digital or physical format).

The challenges that come from using “publishers’ identifiers” throughout the institutional repository framework are fairly obvious:

- Publishers and authors – while part of the same publication workflow – do not typically share the same workflow *system*
- Publishers come later in the workflow than authors – which means attempting to retrofit identifiers to digital objects – inevitably requiring human intervention
- Publishers’ interest in identifying “other editions” is likely to be extremely limited¹³
- Many objects in institutional repositories do not have a “publisher” (aside from the author and/or the repository itself); there are at present, for example, no standard identification schemes for Learning Objects

One other issue perhaps deserves mention here, and that relates to copyright. This is not the place for an extended discussion of the issues relating to copyright and institutional repositories. Nevertheless, it is worth pointing out that whether or not two digital objects are *the same work* from a copyright perspective is potentially a different question from whether they are the same thing from other perspectives. While inconvenient, this cannot be entirely ignored.

2.3 Users

There are, of course, many different potential classes of users of the digital content that might be held in an Institutional Repository. These include:

- Researchers
- Teachers
- Learners
- Information professionals

The extent to which their expectations with respect to identification may differ from one another is not clear to us. Ultimately, all will have similar concerns about understanding the relationships necessary for collocation and disambiguation. They will also face similar challenges with respect to the trust that they need to be able to place in the information provided to them.

¹³ So, while it is possible to imagine an application of the Handle system which resolved the DOI of an article to any one of many different DigitalCopies or DigitalEditions in many different locations, it is hard to imagine this actually being implemented in practice.

3 Scenarios for RIVER report

3.1 Use Case collections

The RIVER Project Team has had access to the following scenarios and Use Cases collections:

Johns Hopkins University project:

A Technology Analysis of Repositories and Services

<https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository>

IMS Global Use Cases

<http://www.imsglobal.org/usecases>

JISC Digital Repositories Programme Scenarios and Use Cases

http://www.ukoln.ac.uk/repositories/digirep/index/Scenarios_and_use_cases

3.2 Scenarios developed by the RIVER Project

The RIVER Project has collected the following scenarios through discussion at the RIVER Workshop in Oxford on 15 February 2006 (see Appendix 2 for attendees), through email discussions and at a project Brainstorming meeting in London on 3 March 2006.

Scenario 1

Type: DigitalVariant

Title: Digital camera image

A photographer takes a photo with a digital camera. The data from the sensor is stored on the memory card in both RAW and JPEG formats. These two digitalVariants are derived from the original image, not from one another. The original itself, the "root version" is non-persistent and cannot be stored.

Scenario 2

Type: DigitalEdition / DigitalEquivalent

Title: Compound object (learning object)

A learning object is created with links to dynamic content on the web. The content is therefore continuously updated and is outside the control of the learning object creator. The question of versioning is irrelevant because the content is changing all the time. In order to express something about the version accessed, learners and teachers need to specify the date on which the content was consulted.

Scenario 3

Type: DigitalEdition

Title: wiki

A research group use a wiki to conduct their project. Documents are added, project communications are carried out through the wiki and the content is dynamically changing. The wiki software allows the researchers and the outside world to track back, using the history function to see the wiki as it was at different points in time. It is also possible to compare the text of documents in the wiki at different stages.

Scenario 4

Type: DigitalEdition

Title: IPR

A thesis submitted to an e-thesis collection exists in two versions. A version with restricted access contains commercially sensitive info. The version made available to the public has had this information deleted.

Scenario 5

Type: DigitalEdition

Title: ArXiv example

A preprint paper is added to the ArXiv subject eprints archive. In due course the paper is refereed and a postprint version is added. The records are linked through the metadata records and through a suffix added to the unique identifier number assigned in ArXiv. The nature of the relationship between versions is indicated by means of a comment. The comment is however placed in a general comments field rather than a specific versions comments field.

Scenario 6

Type: DigitalEdition

Title: Software versions – versions as marketing device

A software package is developed and new developments are issued in managed stages as releases. The move from a version 1.8 to a version 2.0 is agreed as a marketing strategy and an indication that there will be no backwards compatibility from that version onwards.

Scenario 7

Type: DigitalVariant / DigitalEdition

Title: PDF vs HTML format

A document is made available in both PDF and HTML formats. The HTML format is more functional in that hypertext links are enabled in the document whereas they have not been enabled in the PDF. The information payload is the same but the functionality is different.

Scenario 8

Type: DigitalEquivalent / DigitalEdition

Title: Regional edition of a newspaper

A newspaper exists in regional editions. The content varies according to region, but the newspaper is issued under the same title and is considered the same entity.

Scenario 9

Type: DigitalEdition / DigitalCopy / DigitalVariant

Title: Google Scholar search

A search on Google Scholar returns 7 records grouped together. These are a mix of preprint versions, author postprints and published versions. Some of the papers have different content: the preprints are earlier versions than the postprints and published versions. Some of the papers are identical in content and format (preprints posted in PDF format by co-authors at different universities in their institutional repositories). Some are identical in content but different in format: author postprint PDFs taken from a Word document and publisher PDFs

with the journal layout and formatting. The searcher needs to know how many of the 7 papers it is necessary to consult.

Scenario 10

Type: DigitalEquivalent

Title: Learning object with different content (1)

Two Canadian universities use the same learning object for use by their students for the same learning purpose. The content is different in that a culturally relevant content is used at each university.

Scenario 11

Type: DigitalEquivalent

Title: Learning object with different content (2)

University A uses a digital map of Birmingham as part of a learning object. University B uses a digital map of Newcastle to teach the same point. The learning object is functionally the same if the object is considered from the point of view of the learning outcome.

Scenario 12

Type: DigitalEdition

Title: Private personal resource management

A university repository is established with four functional areas: a private area for early versions submitted by and accessible to individuals, a collaborative area for private but team use, a restricted layer made available to university members only, and a public area. An author will revise documents and at each stage of revision they may be made accessible in a different area of the repository in accordance with the author's wishes.

Scenario 13

Type: DigitalVariant / DigitalEdition

Title: Best version varies depending on identity of searcher

Researcher A searches for an article by Professor X and retrieves three versions: a preprint, an author postprint and a publisher version. She wishes to cite the article and to be sure that she reads the same version as the one she is citing. Her university subscribes to the journal so she is able to access the best version for her.

Researcher B searches for the same article and retrieves the same three versions. He is a freelance journalist without straightforward access to subscribed resources, but he needs to access the article. He reads the postprint version and cites the research. The best version for him is the latest open access copy.

Scenario 14

Type: DigitalEdition / DigitalCopy

Title: Trust in open access versions

For the Research Assessment Exercise (RAE), one of the panels have indicated that they do not feel able to trust open access copies of author versions. The concern is that the open access versions may have been tampered with, eg by authors erasing citations in their articles to make them look less derivative. For the RAE panel the publisher versions are the ones they can trust, though they would trust open access versions if they could verify that the information payload is the same.

Scenario 15

Type: DigitalEdition

Title: Public versions static

An author posts a version of their work in an institutional repository and according to the university repository policy, it must remain static once made public. The author wishes to make a change to the document. This is not permitted according to the universities policy.

An author at another university wishes to amend a paper they have posted. This is granted by a request to the repository manager and the change is reflected in the metadata record.

Scenario 16

Type: DigitalEdition

Title: Versioning of private versions

A university supports its researchers by providing personal resource management tools. A simple date-time stamp differentiates between different private versions of the document. Documents may be amended and removed while still in their non-public phase.

Scenario 17

Type: DigitalEdition

Title: Researcher moves universities

A researcher deposits all their preprints in University Theta. They then move institutions and deposit all their preprints in University Omega. They have changed one of the preprints but do not amend the metadata. Researchers will generally remain unaware of the fact that these papers do not have identical content.

Scenario 18

Type: DigitalEdition

Title: Re-use of data file

A researcher deposits an Excel dataset. Another researcher takes the file, re-uses it and deposits it but the data has changed. It is very hard to spot the changes without a close comparison of the two files, and the second version may be "better" than the first. It also may be the case that the second version overwrites the first.

Scenario 19

Type: DigitalVariant / DigitalEdition

Title: TEI format

A researcher produces a Word document and then enriches this by adding coding and mark up using TEI. The information payload is the same, but the coding adds greatly to the functionality of the document.

Scenario 20

Type: DigitalRevision / DigitalEdition

Title: Federated search

A researcher looks for a paper in OAISter and finds two entries, one listed as an article and the other as a preprint. The metadata does not indicate date of issue of the two entries or any other detail about what the differences may be between

the two versions. The researcher has to track and view both versions and make a visual comparison.

Scenario 21

Type: DigitalEdition

Title: Use of word processor metadata capabilities

An author creates a document using Word or Open Office. They complete the simple metadata and version control fields provided by the software. On converting the document to a PDF file for deposit in their institutional repository, the author notes that these metadata elements have been lost.

Scenario 22

Type: DigitalVariant

Title: Recordings of performances

A digital recording is made of a dance performance along with a high quality audio-only recording. In addition several still images of the performance are captured. These are all deposited in an institutional repository and for the searcher's purpose they are versions of the same thing: the performance itself.

Scenario 23

Type: DigitalCopy

Title: Archive version

University X applies a policy of holding an archive version of each digital object held in the repository. This version does not get touched as it is the authority version.

Scenario 24

Type: DigitalEquivalent

Title: Eiffel Tower

A picture library holds multiple images of famous monuments, such as the Eiffel Tower. These are considered to fulfil the same function as each other and the images are collocated through the metadata. Are these versions of each other in any sense?

Scenario 25

Type: DigitalEdition

Title: Best version identified through use

An open source software community is active in using a particular version of the software, not the very latest version. This is because the latest version is particularly buggy and so has not been widely adopted. Development and use of the software is taking place on an earlier version which is more reliable.

Scenario 26

Type: DigitalCopy / DigitalEdition

Title: Subject repositories vs Institutional repositories

Researchers at university Z deposit their papers in both a subject repository such as ArXiv or PubMedCentral and in an institutional repository. The local copy may be downloaded by the university from the subject repository or researchers may be requested to deposit their paper in both repositories. Which is the "official" version?

Scenario 27

Type: DigitalEdition

Title: Provenance

A full versioning statement allows users to go back in time to what an object was like at a particular stage, working back. There is an analogy with a painting and the preliminary sketches made in preparation for it.

Scenario 28

Type: DigitalEdition

Title: Magazines and legal challenges

A magazine's editorial processes allow for versions to be held in workflow so that any legal challenge such as libel can be traced back to source.

Scenario 29

Type: Digital Revision / DigitalEdition

Title: What is the difference?

A researcher finds multiple versions of a paper and wishes to know whether there are substantive differences. If not, she will be happy with the latest version. If there are significant differences she would like to have a quick way of knowing this, so she can make a judgment about whether to refer back to earlier versions.

Scenario 30

Type: DigitalCopy / DigitalEdition

Title: Very large number of copies in different locations

A particular physics paper has 47 authors in as many institutions. Each author deposits their copy in their institutional repository. How can the searcher deal with this multiplicity of copies and differentiate between them?

Scenario 31

Type: DigitalEdition

Title: Citation of instance

A researcher is reading an open access copy of a paper stored in his institutional repository. There is a difference between the paper and the final published article, of which the researcher is aware. He cites the paper giving both the publisher citation and the reference to the repository instance, and indicating that it is the open access version he has consulted.

Scenario 32

Type: DigitalVariant

Title: Postprint has same information payload as published version

An author deposits a postprint in her university institutional repository. There were no last minute corrections at the proof-reading stage, so she knows that the postprint is the same as the published article in all but layout. She asks the repository staff to indicate this in the metadata record and on the cover sheet which they are adding to all papers.

Scenario 33

Type: DigitalVariant

Title: Thumbnails

A repository administrator is adding images to a repository. For each digital object he adds the image in thumbnail format as well as the full preservation

copy version. The photographer has also submitted zooms and photograph sections to be added as separate datastreams.

Scenario 34

Type: DigitalEquivalent

Title: VLE

A teacher adds an object to the university Virtual Learning Environment (VLE). There is a dialogue going on with the learners, with ongoing updating processes, content that includes annotations, discussions, and reading lists. The concept of versions of the object may not be relevant here.

Scenario 35

Type: DigitalEdition

Title: Deleted versions

A paper has been submitted to an institutional repository, but has to be withdrawn for legal reasons. The record is deleted by the repository staff. The institutional repository policy is to support deleted items at the OAI-PMH "persistent" level, therefore harvesters will pick up on the deletion. A later version still exists in the repository and it would be useful for searchers to be directed to this when searching for the deleted version.

Scenario 36

Type: DigitalEdition

Title: Translation

An author has published an article both in English and in Chinese. They wish to deposit open access copies in both language versions in the institutional repository and to have the records linked so that searchers can easily find them.

3.3 Use cases submitted to the RIVER workshop on 15 Feb 2006

Use Case 1

A paper is written by four co-authors: A, B, C and D. Authors A and B come from the same institution (I), author C comes from institution J and author D from institution K. Author A deposits the journal-submitted preprint in his institutional repository.

Author B doesn't realise that A has already deposited a preprint and deposits an author postprint into the same repository in a separate record. Author C, the lead author, deposits the publishers' PDF into her institutional repository.

Author D, doesn't bother depositing the full text, but creates an 'empty' eprint record in his repository. However, the metadata that D creates is more complete than any of the other records, including the journal ISSN and a complete list of references.

How would an RAE-aware librarian from I, J or K find and choose **the appropriate version** to link to for their institutional RAE submission?

Use Case 2

A bibliographic citation discovery service wants to provide a link for a user to a discovered article within an open access repository when the user does not have

a subscription to access the published version of the article and doesn't have an OpenURL resolver available. Also it would like to provide links to conference papers, the full text of which is less easy to locate. Thus the service wants to locate an article or paper using its bibliographic citation details. It will have an additional preference to locate the **published version** of the paper rather than any drafts or preprints, but it may accept **an earlier version** if that is the only one available within the repository.

Proposed solution: data matching harvested records

Requirements: universally implemented conventions for "publication status" syntax and semantics within OAI-PMH records

Use Case 3

- A researcher finds multiple open access versions of an academic paper and wants to identify quickly **the "best" version** or versions. The researcher wants to be aware of and have access to the published version if this exists and wants to understand the differences between different versions.
- The researcher searches for academic paper via Internet search engine or through open access service provider
- Repository or service returns multiple versions of the same article
- Service provider presents results in such a way that researcher understands the status of versions and their relationships to each other
- Researcher reads the best version or versions

Proposed solutions

- hierarchical linking of and display of metadata records
- specific provision for free text remarks about relationships in both metadata and cover sheet of the document
- some technical solution to textual comparison (involves storing the text in other formats in addition to PDF)

3.4 Summary of requirements from scenarios and use cases

The requirements listed in Table 5 below are drawn from the scenarios and edge cases listed above and are addressed to repository software developers, metadata standards community, repository policy makers, and repository managers.

Table 5. A summary of requirements drawn from the scenarios gathered by the RIVER project team.

Requirement ID number	Category	Requirement	Scenario number (s)
R1	Disambiguation	Differentiate between digitalCopies, digitalRevisions, digitalEditions, digitalVariants	9, 29, 30

Requirement ID number	Category	Requirement	Scenario number (s)
R2	Disambiguation	Use date and time information, textual comments and schemas as minimum methods of disambiguation	13, 20
R3	Disambiguation	Disambiguate objects with the same information payload but different formats and functionality	7, 19, 33
R4	Disambiguation	Describe the nature of the relationship between digital objects – in human readable form	1, 4, 5, 9, 22, 29, 36
R5	Disambiguation	Describe the nature of the relationship between digital objects – in a way that can be used in machine-to-machine processes	1, 4, 5, 9, 22, 29, 36
R6	Disambiguation	Support criteria other than date for identifying the “best” instance, eg format, access level	25, 26
R7	Disambiguation	Provide explicit indication if an object is dynamic / continuously updated or if it links to external dynamic content	2, 3, 34
R8	Disambiguation	Provide recommended citation style for open access objects, including recommended citation of objects with dynamic content	3, 31
R9	Collocation	Consider function for the user as an important attribute in collocation of digitalEquivalents, even where content or information payload may differ	10, 11, 24
R10	Collocation	Enable the description of digital objects to include links to related objects and for these links to be used by search services	1, 4, 5, 9, 22, 33, 36
R11	Collocation	Enable links to published versions where available using persistent identifiers	13, 31
R12	Collocation	Express relationships to objects that are absent (not stored permanently or stored outside the repository)	1, 35
R13	Collocation and Disambiguation	Work with search engine and other search services to improve collocation and disambiguation outside digital repositories	20
R14	Collocation and Disambiguation	Implement versioning information in OAI-PMH to ensure that information recorded in digital repositories is picked up by federated search and Internet search engines.	9,17,20,26,30
R15	Revision control	Manage different access levels for digital objects during revision process, eg personal, collaborative, restricted, public	12
R16	Revision control	Control objects once placed in a public state in a digital repository to ensure that content does not change without this being reflected in the metadata record. Apply process/ workflow NNN	14, 15, 32

Requirement ID number	Category	Requirement	Scenario number (s)
R17	Revision control	Allow for revision and deletion of private versions of objects stored in the repository – personal resource management	16
R18	Revision control	Maintain archive versions of each digital object which are kept as authority versions and are not touched after deposit	23
R19	Revision control	Provide tools for comparison of content of open access objects with each other and with published versions, eg textual comparison	13, 14, 17, 18, 29, 32
R20	Revision control	Provide tools for historical comparison within an object, along lines of CVS and wiki	27, 28
R21	Workflow	Capture simple metadata created by authors, eg from word processed files	21
R22	Workflow	Provide support and workflow for authors who wish to deposit objects in multiple repositories/locations multiple deposit of objects	26
R23	Workflow	Promote the use of deposit APIs which support version control	12, 16, 27, 35

4 Review of current repository practice

4.1 Introduction

Version control in the creation and management of digital objects often starts at the desktop with an informal form of version control using custom file names, version numbers etc.¹⁴ Only when the author (or some other party) considers the object to have reached an appropriate version does it get deposited within a formal repository. For many digital objects that is the perceived end of the process. For others, however, changes may continue to be made or derivatives made. Within the formal setting of an institutional or subject repository are there any processes in place to formalise the management of multiple versions? Do the policies and systems handle changes or generation of different versions as a part of the process of managing digital objects within the repository? If formal version control is to start earlier in the process then there is a real need for a tight coupling between the authoring tool and the repository service or filestore, or at least an accepted workflow which guides authors through a controlled submission process, including the issuing of meaningful identifiers. There is a predictable convergence between repository systems and the detailed version control and history found within wiki environments (which itself is derived from CVS for software development).

Repositories are used in conjunction with a wide range of tools: word processors, diagramming, scanners, photocopiers, search engines, email, image capture, statistical analysis software and so on. In turn people adopt many roles to enable the collaboration that is necessary to move from the inception of an academic idea to a published work. The workflows inferred by this complex engagement between tools and people are diverse and complex.

Metadata is created to enable resources to be managed and located. There are two main ways that metadata within a repository is shared with other systems: through allowing web robots to harvest the metadata or allowing a federated search query to retrieve a result set. Whatever the mechanism, a potentially large set of metadata is presented to a user in relation to the search term they are researching and the user needs to be able to scan this metadata set to find the resources that will serve their needs. It needs to be easy for the user to ascertain and comprehend useful information about the version, revision and access rights associated with a resource.

4.2 Methodology

Investigations into current approaches to identification and version control of digital objects within repositories began with a sweeping of the sector, in particular using the OpenDOAR registry (<http://www.opendoar.org/>). Whilst the OpenDOAR registry is a useful starting point, it is worth noting that OpenDOAR concentrates on: a) open access repositories; b) research repositories (whether institutional or not); c) only an indication of content type can be provided, relying on, we think, the publication types declared by the repository (without necessarily any semantic agreement on what those publication types are). OpenDOAR does not, as yet, capture any information about how, or if, the repository identifies objects or controls versioning.

14 The RepoMMan project (University of Hull) analysis of requirements suggests that easy access from within office applications together with a usable document sharing and versioning facility should be part of the set of requirements for an institutional repository.

For the purposes of this exercise we were interested in having a broad definition of “repository” in order to compare the use of identifiers and version control across a range of systems (some of which are emerging through the so-called “social computing” movement). Thus, we have included both the archetypal CVS and the emergent wiki within our spectrum. The Dictate Project¹⁵ which enhances the EPrints software to enable user “tagging” suggests that software to support repositories and software to support social computing will converge. However, no repository has yet implemented a version control system anything like that found in most wiki implementations, for example.

The following sections focus on current practice across repositories. We have included a series of case studies which focus on both specific repository instances (the totality of software, policy and service) and on the more common repository software systems (e.g. eprints.org and DSpace).¹⁶ In summary, the review of practice focuses on the following areas:

- Documenting the identifier and versioning practice of major repository services;
- Documenting the technical facilities for identifier management and versioning provided by the underlying repository systems (where these are available);
- Results from an analysis of user interfaces, technical documentation and consultation with repository managers.

4.3 Repositories and content types

The following are examples of content found within institutional and subject repositories. We have divided them by content typically identified by file format and that identified by type of publication.

- Types identified by file format:
 - word processed document: doc, rtf, TeX
 - spreadsheet
 - database
 - source code
 - zip
 - image
 - movie
- Types identified by specified publication type:
 - preprint
 - postprint
 - thesis
 - book
 - book chapter
 - course material
 - assessment

15 http://www.jisc.ac.uk/index.cfm?name=project_dictate

16 An excellent overview and comparison of the available repository software solutions is provided by A Guide to Institutional Repository Software (3rd ed). Open Society Institute, 2004.
<http://www.soros.org/openaccess/software/>.

- learning object
- presentation

These do not, of course, have fixed boundaries but the approach serves to indicate that there is more than one way to identify content-types within a repository. Not surprisingly the bulk of content within repositories relates to research outputs (e.g. eprints, theses, monographs, and to a lesser extent presentations and research data). For the most part these objects are delivered as PDF files (though potentially they may be ingested in another format, e.g. TeX, Microsoft Word).

4.4 Case studies

Table 5. Case studies showing current practices across repositories

Repository	ArXiv ¹⁷
Web address	http://arxiv.org
Underlying software solution	Proprietary solution not available separately.
What types of objects is the repository used to store?	<i>Formats:</i> ArXiv prefers TeX (from which other formats such as PDF are derived). Failing that, authors can submit postscript or PDF. <i>Publication types:</i> pre-prints, post-prints (as noted by OpenDOAR) but also theses (e.g. http://arxiv.org/abs/hep-lat/0508002) and conference papers. A number of records have reference to supporting powerpoint or other materials which sit outside arXiv (e.g. http://arxiv.org/abs/astro-ph/0405196).
Which communities predominantly use the repository?	Research: Physics, Mathematics, Computer science, Non-linear sciences, Quantitative biology.
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	For each object there is one metadata record (e.g. http://arxiv.org/abs/cs.DL/0504084). The metadata record provides links to previous or subsequent versions (each of which is treated as a separate object with its own metadata record).
How are revisions identified?	Versions are identified by suffixing vn (where n is the version number) to the identifier (e.g. http://arxiv.org/abs/cs/0504084v3). It is up to the author(s) to include a comment summarising the differences between versions.
What facility does the repository provide to associate different versions?	ArXiv speaks of “replacing” a paper. The service provides a “replace” form to complete which will associate the replacement with the paper’s identifier. Replacements generate a new version as described above. The only exception is replacements submitted within the same day (before 16:00 US Eastern time (EDT/EST) Mon-Fri). It is a policy decision that previous versions of a paper remain available in order to help “archive the historical record of research”.

¹⁷ Arxiv was selected because it is the longest-established open access eprints repository, having been established in 1991. This case study provides a benchmark for policy and practice. See, e.g., Richard E. Luce, “E-prints Intersect the Digital Library: Inside the Los Alamos arXiv”. *Issues in Science and Technology Librarianship* (Winter 2001): <http://www.istl.org/01-winter/article3.html>

Repository	ArXiv ¹⁷
How are resources described in the system i.e. what facilities for metadata management are provided?	ArXiv uses a series of web forms to collect the metadata and upload the file(s). Replacing a paper allows the editing of metadata fields (but as a new version)
Is metadata in the repository exposed to search engines and OAI harvesters?	ArXiv is a registered OAI service provider. ArXiv supports OpenURL and RSS Arxiv content is fully indexed by Google Scholar

Repository	DSpace @ Cambridge
Web address	http://www.dspace.cam.ac.uk/
Underlying software solution	DSpace ¹⁸
What types of objects is the repository used to store?	Preprints, postprints, monographs, working papers, multimedia, web pages, grey material (not all materials are open access)
Which communities predominantly use the repository?	Cambridge research community
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	DSpace History System uses ABC Harmony data model. A high-level critical analysis of the History System has been published as part of the PLEDGE Project (http://simile.mit.edu/pledge/HistoryRecap and http://wiki.dspace.org/HistorySystem). ¹⁹
How are revisions identified?	Not known, and possibly not possible save within the metadata description.provenance or similar
What facility does the repository provide to associate different versions?	Whilst it was difficult to discover objects with branching or linear items within Cambridge DSpace, the MIT DSpace instance included branched versions (e.g. public and private versions) as files within the same overall object. In theory, a DSpace instance could make use of locally qualified Dublin Core metadata to make statements about the relationships between objects (e.g. relation.isformatof; relation.ispartof; relation.ispartofseries; relation.haspart; relation.isversionof; relation.hasversion; relation.isbasedon). In the time permitted it has not been possible to find specific examples of this usage in practice.
Is metadata in the repository exposed to search engines and OAI harvesters?	DSpace supports OAI.

¹⁸ <http://www.dspace.org/>

¹⁹ See also <http://wiki.dspace.org/VersioningSupport> which includes an outline plan for displaying relationships between multiple versions of an object in DSpace.

Repository	ARROW (Australian Research Repositories Online to the World) ²⁰
Underlying software solution	Fedora provides the backend, common repository and (with VITAL) the middleware to support the range of publication types. Fedora is not an “out of the box” repository service but rather provides the storage layer. Other software components are used for e.g. the ingest and presentation interfaces. VITAL provides the presentation interfaces.
What types of objects is the repository used to store?	Preprints, postprints, monographs, working papers, theses etc
Which communities predominantly use the repository?	Australian research community
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	<ul style="list-style-type: none"> ▪ Each object comprises a metadata set and one or more data streams. ▪ The CNRI handle system is the chosen persistent identifier method. ▪ Handles are assigned to each data stream or component of a digital object ▪ Fedora provides version control through automatic storage of content and metadata when modified ▪ Fedora enables retrieval of objects based on date-time stamps.²¹
What facility does the repository provide to associate different versions?	It has not been possible to find examples of branched or linear versions in practice.
How are resources described in the system i.e. what facilities for metadata management are provided?	Clients for Fedora are developed separately. Fedora provides the storage layer. UNSW have deployed the VTLs Valet component which provides a web interface for submission of electronic theses and other digital objects (see http://www.fedora.info/tools/).
Is metadata in the repository exposed to search engines and OAI harvesters?	Fedora supports OAI out of the box.

Repository	ePrints Soton
Web address	http://eprints.soton.ac.uk/
Underlying software solution	EPrints software (http://www.eprints.org/software/)
What types of objects is the repository used to store?	Documents e.g. word and pdf
Which communities predominantly use the repository?	Research communities
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	As web addresses or URLs

²⁰ ARROW (<http://arrow.edu.au/>) is a consortium project funded by the Australian Commonwealth Department of Education, Science and Training (DEST) and is implementing software to support institutional research repositories for eprints, theses and electronic publishing. ARROW have implemented Fedora together with VTLs VITAL.

²¹ Version control in Fedora is documented in “Fedora Content Versioning”.
<http://www.fedora.info/download/2.1/userdocs/server/features/versioning.html>.

Repository	ePrints Soton
How are revisions identified?	Through appending an id onto a URI where the id is generated from a counter that starts as 00000000 within a repository instance e.g. Example ID: http://eprints.soton.ac.uk/5950/
What facility does the repository provide to associate different versions?	A web page ("splash" page) can be used to store related works
How are resources described in the system i.e. what facilities for metadata management are provided?	<ul style="list-style-type: none"> ▪ Simple Dublin core ▪ New system (released end of March as Alpha version) is providing facilities to describe resources with METS and DIDL although Southampton team is awaiting metadata aggregators to tell them how these schemas should be used
Is metadata in the repository exposed to search engines and OAI harvesters?	Yes, e.g. OAI id for that record: oai:eprints.soton.ac.uk:5950

Repository	JORUM
Web address	http://www.jorum.ac.uk/
Underlying software solution	Intralibrary: http://www.intrallect.com/
What types of objects is the repository used to store?	IMS Content Packages and associated resources e.g. documents and images
Which communities predominantly use the repository?	Learning and teaching
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	Internally Jorum can store and manage multiple versions of a resource. Once "published" the unique URL is fixed, e.g. http://repository.jorum.ac.uk/intralibrary/IntraLibrary?comm=and=open-preview&learning_object_key=i2886n34968t http://repository.jorum.ac.uk/intralibrary/virtual_file_path/227/unit5/q0.html
How are revisions identified	As separate resources with their own UID. Revisions cannot be expressed publicly
What facility does the repository provide to associate different versions?	None
How are resources described in the system i.e. what facilities for metadata management are provided?	UK LOM Core records where author can edit the version field
Is metadata in the repository exposed to search engines and OAI harvesters?	Yes

Repository	Mediawiki at Oxford
Web address	http://ask.oucs.ox.ac.uk/ask/
Underlying software solution	Mediawiki is one of a range of software artefacts that fall under the general heading of wikis. Mediawiki: http://www.mediawiki.org/wiki/MediaWiki MoinMoin: http://moinmoin.wikiwikiweb.de/

Repository	Mediawiki at Oxford
	Oxford University is using mediawiki for private, public and collaborative work. MediaWiki is the underlying software that the Wikipedia projects use. Wikipedia is one of the most popular sites on the internet.
What types of objects is the repository used to store?	Text and images
Which communities predominantly use the repository?	At Oxford University Computing Service, wikis are being used to document internal process, augment the department website and to aid collaboration with other organisations There is a growing interest in wikis in academic departments too.
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	Full version control although wikis normally lack locking mechanisms
How are revisions identified	By the author creating new pages or subsections in a page
What facility does the repository provide to associate different versions?	By structuring the wiki as desired
How are resources described in the system i.e. what facilities for metadata management are provided?	Wikis don't normally provide structured metadata entry facilities
Is metadata in the repository exposed to search engines and OAI harvesters?	No

Repository	SVN (Sourceforge)
Web address	http://sourceforge.net
Underlying software solution	Oxford uses SVN that is hosted both locally and for global use e.g. by Sourceforge http://sourceforge.net/docman/display_doc.php?docid=31070&group_id=1
What types of objects is the repository used to store?	Source code e.g. C++, Java
Which communities predominantly use the repository?	Developers
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	Full metadata and object version identification available over binary and metadata CVS repositories are not normally public so public URLs are not generated Exception is new Sourceforge CVS facility
How are revisions identified	Full change control history and locking to prevent concurrent authoring
What facility does the repository provide to associate different versions?	New "branches" in the code tree which can normally be visually inspected

Repository		SVN (Sourceforge)
How are resources described in the system i.e. what facilities for metadata management are provided?	None. If the user wants to find a specific code resource they would need to move the code to an Integrated Development Environment (IDE) such as Eclipse and then use the Search facilities there. Or the user could search say Javadoc HTML if this has been created	
Is metadata in the repository exposed to search engines and OAI harvesters?	No	

Repository		TOIA at the University of Strathclyde
Web address	http://www.toia.ac.uk/	
Underlying software solution	IIS and IE5	
What types of objects is the repository used to store?	Assessments Assessment questions	
Which communities predominantly use the repository?	Learners and teachers	
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	A new version is simply a copy of an existing resource. It may be associated (loosely) by a search service if it had similar metadata	
How are revisions identified	TOIA provides a simple replace mechanism, the user simply updates an assessment or question. Each resource is given an internal GUID e.g. 41105ac0-2327-4978-8178-bf4b547ffbfc Only assessment are given public URLs (i.e. not individual questions) and these are generated according to the schematic: http://<Host address>/AMSTOIAPS/ frmPSLogin.aspx?ln=admin&ufn=a&uem=a&aid=1 ln=Login name of user ufn = first name of user uem= user email address aid= assessment id to present	
What facility does the repository provide to associate different versions?	None specifically for this purpose although the hierarchical structures that can be created within the system could be used to associate versions but this would be represented for consumption outside of the system	
How are resources described in the system i.e. what facilities for metadata management are provided?	All assessments and questions are described using the IMS metadata schema The system does not let the author enter version number	
Is metadata in the repository exposed to search engines and OAI harvesters?	No . Recent developments mean a prototype SRW interface is exposed so that federated TOIA instances can be cross searched	

Repository		ARNO
Web address	http://www.uba.uva.nl/arno	
Underlying software solution	ARNO software. http://arno.uvt.nl/~arno/site/index.html	

Repository	ARNO
What types of objects is the repository used to store?	The ARNO (Academic Research in the Netherlands Online) Project is developing and implementing document servers at six Dutch institutions to enable exposure of scientific outputs. The Universiteit Maastricht runs the ARNO software and document types include articles, books, chapters, lectures, and reviews. Not all metadata records give access to the fulltext, however. The Universiteit Twente repository which is running on i-Tor (http://www.i-tor.org/en/system_info/about/) contains a mixture of articles, theses, editorials, books and book chapters.
Which communities predominantly use the repository?	The ARNO repositories aim to make available the research outputs of Dutch universities. The user community is likely to be primarily members of the international research community.
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	Each document receives a unique internal system ID. For each document there a corresponding metadata record (though it is not essential for metadata records to have corresponding fulltext documents). Metadata records are linked via "previous" and "next" version fields.
How are revisions identified	Arno does not explicitly differentiate between revisions and versions. There appears to be no obvious way of declaring how one version/revision might differ from another.
What facility does the repository provide to associate different versions?	Within a metadata record fields exist to specify the "previous version", the "date the record changed" and the "next version", together with the name of the user(s) making the change. Within a metadata section relating to "release information" it is possible to explicitly specify a version code for the document.
How are resources described in the system i.e. what facilities for metadata management are provided?	ARNO provides a metadata management screen for depositors and for archive/repository editors. Bibliographic metadata is collected together with data relating to the status and document "header".
Is metadata in the repository exposed to search engines and OAI harvesters?	ARNO is an OAI data provider and captured metadata is transformed to Dublin Core for this purpose. ARNO also provides a Z39.50 service.

Repository	CERN Document Server
Web address	http://cds.cern.ch/
Underlying software solution	CERN Document Server Software (CDSWare) -- http://cdsware.cern.ch/
What types of objects is the repository used to store?	Articles/preprints; books and proceedings; lecture and presentations; reports; multimedia; and archives. Of the 833,000 records approximately 360,000 have fulltext associated with them.
Which communities predominantly use the repository?	Research community
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	An internal identifier is allocated to the metadata record. The format is similar to ArXiv and the document name is derived from the record identifier (plus an appropriate extension, depending on the format). Like ArXiv, version numbers are appended to the document identifier.
How are revisions identified	No distinction is made between revision and version. "Revised versions" are uploaded and attached to a single metadata record.

Repository	CERN Document Server
What facility does the repository provide to associate different versions?	One metadata record points to one or more versions of the document.
How are resources described in the system i.e. what facilities for metadata management are provided?	Editing forms capture bibliographic information. Internally the record structure complies with MARC21. Data can also be bulk uploaded from OAI repositories, bibliographic catalogues etc.
Is metadata in the repository exposed to search engines and OAI harvesters?	CDSWare provides an OAI data provider.

Repository	PubMedCentral
Web address	http://www.pubmedcentral.nih.gov/
Underlying software solution	Portable PMC (pPMC) is a searchable online repository of medical journal articles. Installed locally it will provide a web application for searching a MS SQL database and rendering; synchronization utilities. Currently searches are sent to the PMC but fulltext is retrieved locally. Future versions will permit local indexes and the retrieval of content not available within PMC.
What types of objects is the repository used to store?	Journal articles and author manuscripts.
Which communities predominantly use the repository?	Research community in biomedical sciences.
How are versions of (a) metadata and (b) objects/binaries identified in the repository?	<p>Articles are allocated a PMC ID (e.g. PMID: 340090). For each fulltext article a Pubmed metadata record exists. Within the metadata record other IDs are listed where they exist. For example:</p> <pre><ArticleIdList> <ArticleId IdType="pii">1BGEBDAJHQFVMWTG</ArticleId> <ArticleId IdType="doi">10.1098/rsta.2005.1610</ArticleId> <ArticleId IdType="pubmed">16099751</ArticleId> </ArticleIdList></pre> <p>A large proportion of content in PMC is supplied by the publisher and tends to be the final copy. Content is encoded in XML. The National Center for Biotechnology Information (NCBI) of the National Library of Medicine (NLM) have created the Journal Archiving and Interchange Document Type Definition (DTD) suite (http://dtd.nlm.nih.gov/). Publishers are encouraged to use the Journal Publishing DTD/Schema (http://dtd.nlm.nih.gov/publishing/). Authors of articles in journals not included within PMC are encouraged to deposit "the final, peer reviewed manuscripts of such articles once they have been accepted for publication" and may submit in standard office application formats.</p>
How are revisions identified	<p>This is possibly not applicable. The Pubmed metadata record about a fulltext article can include a <history> element which includes within it data about the publisher's processing history of the document. (e.g. change of status). Child elements comprise pairs of data and text fields. Example:</p> <pre><History> <PubMedPubDate PubStatus="received"></pre>

Repository	PubMedCentral
	<pre> <Year>2005</Year> <Month>6</Month> <Day>22</Day> </PubMedPubDate> <PubMedPubDate PubStatus="revised"> <Year>2005</Year> <Month>12</Month> <Day>15</Day> </PubMedPubDate> <PubMedPubDate PubStatus="pubmed"> <Year>2006</Year> <Month>3</Month> <Day>8</Day> <Hour>9</Hour> <Minute>0</Minute> </PubMedPubDate> <PubMedPubDate PubStatus="medline"> <Year>2006</Year> <Month>3</Month> <Day>8</Day> <Hour>9</Hour> <Minute>0</Minute> </PubMedPubDate> </History> </pre>
What facility does the repository provide to associate different versions?	<p>The Archiving and Interchange DTD includes the element, "<code><related-article></code>" for "related article information". This element, which the documentation admits is somewhat overloaded, is intended to express a relationship to one or more related articles within the encoded metadata of a journal article; and as a general linking element which can describe many types of relationship from within the text to another article. The <code><related-article></code> appears to be only means of associated different versions of an article. The element can include one or more attributes. Relevant attributes include: "alternate-form-of" and the mandatory "related-article-type". Suggested values for the latter attribute include: addendum, companion, corrected-article, correction-forward, republished-article, retracted-article, retraction-forward.</p>
How are resources described in the system i.e. what facilities for metadata management are provided?	<p>Publishers submit content as XML. Authors complete a simple form which collects basic details about the publication and enables upload of the manuscript, including any supplementary data.²² Details of the internal metadata management systems are not known.</p>
Is metadata in the repository exposed to search engines and OAI harvesters?	<p>Support for OAI 2.0 is included. A series of Entrez tools are available to enable machine-2-machine calls to the central databases from remote clients.</p>
Note:	<p>The UKPMC Implementation Group, led by the Wellcome Trust, are procuring a UK version of PMC, including use of the pPMC software – see further http://www.wellcome.ac.uk/doc_wtd015366.html</p>

22 The Wellcome Trust provides a short guide to submitting a manuscript to Pubmed Central together with an animated tutorial, http://www.wellcome.ac.uk/doc_WTD018855.html#P98_9145.

4.5 Identity and version management in the OAI Protocol (short note)

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH²³) is a lightweight protocol for exposing and harvesting metadata about digital objects stored in digital repositories. Most institutional repository systems now offer an OAI data provider service which enables an OAI harvesting service to retrieve one or more metadata collections from which aggregation and search services may be built.²⁴

The OAI-PMH assumes a repository contains items from which XML metadata records can be generated and for each a (locally) unique identifier exists. OAI metadata can be expressed in a variety of formats. The default, mandatory format for interoperability is simple Dublin Core. The unique identifier for any given item remains the same regardless of the format in which the metadata record is written. The combination of unique identifier and datestamp allows for selective harvesting (e.g. to test for new, modified or deleted records since or within a given date range). A simple Dublin Core metadata record will contain an identifier element, the content of which will point to the digital object being described (e.g. fulltext item, or richer finding aid).

Dublin Core also espouses the "one to one" principle, that, "Dublin Core metadata describes one manifestation or version of a resource, rather than assuming that manifestations stand in for one another. For instance, a jpeg image of the Mona Lisa has much in common with the original painting, but it is not the same as the painting."²⁵

Whilst Dublin Core has a "relation" element, without the use of qualifying terms, its use may be too ambiguous for relating versions of an item to each other (though the content of the element might point to an unambiguous relationship between two objects). In practice, OAI Dublin Core metadata records tend to describe an overall summary or entry page for an "item". The opening page may point to more than one version or manifestation of the item (e.g. different presentation formats or backward/foreword versions).

Example OAI record (courtesy of ePrints Soton, <http://eprints.soton.ac.uk/15759/>). The <relation> element is used in this example to point to a revised version of the fulltext.

```
<record>
  <header>
    <identifier>oai:eprints.soton.ac.uk:15759</identifier>
    <datestamp>2005-05-27</datestamp>
    <setSpec>7374617475733D7375626D6974746564</setSpec>
```

23 OAI-PMH 2.0, <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>

24 A list of registered OAI data providers is available at <http://www.openarchives.org/Register/BrowseSites>. OAIster is an example of an OAI harvesting service which provides aggregation and search, <http://oaister.umdl.umich.edu/o/oaister/>.

25 Diane Hillmann, "Using Dublin Core" (2005). <http://dublincore.org/documents/usageguide/#whatis>

```
<setSpec>7375626A656374733D47:4745</setSpec>
<setSpec>7375626A656374733D47:4743</setSpec>
<setSpec>67726F75703D756F732D686B</setSpec></header>
<metadata>
  <oai_dc:dc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd"
xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>A method for tracking individual planetary waves in remotely sensed
data</dc:title>
    <dc:creator>Cipollini, P.</dc:creator>
    <dc:creator>Challenor, P.G.</dc:creator>
    <dc:creator>Colombo, S.</dc:creator>
    <dc:subject>GE Environmental Sciences</dc:subject>
    <dc:subject>GC Oceanography</dc:subject>
    <dc:description>We describe a methodology for tracking individual planetary
waves in longitude-time plots of satellite data [...]</dc:description>
    <dc:publisher>Southampton Oceanography Centre</dc:publisher>
    <dc:date>2005-05-19</dc:date>
    <dc:type>Article</dc:type>
    <dc:type>PeerReviewed</dc:type>
    <dc:identifier>http://eprints.soton.ac.uk/15759/</dc:identifier>
    <dc:format>application/pdf</dc:format>

    <dc:relation>http://eprints.soton.ac.uk/15759/01/Cipollini_et_al_revised.p
df</dc:relation></oai_dc:dc></metadata></record>
```

5 Mechanisms we have found in use for fulfilling the identified requirements

5.1 Introduction

There are many different ways in which it is possible to approach the implementation of solutions to the requirements outlined in Table 5 in Section 4 above. However, there are very few standardised mechanisms in use within the repositories which we sampled which would suggest mechanisms for achieving interoperability between the range of repository solutions.

5.2 Disambiguation and collocation

- For disambiguation and collocation *within a single repository* Many repositories employ the use of a “splash” web page to aggregate related resources; the effectiveness of this depends on either the depositor (or a repository manager) controlling deposit.
- It is unclear whether it should be the responsibility of a repository to maintain a record of all the network locations of digital copies of resources. Such a service would seem to be best abstracted as a distinct shared service registry.
- More loosely search engines can address collocation by associate resources simply because their metadata is similar so they would be presented to the user who can visually scan and spot the similarity (collocation) by sight; this is however not a scalable solution,
- Resolvers can be configured to provide links to a range of digital copies at different network locations (thus allowing the user to discern the best or appropriate copy) but only if there is a mechanism for disambiguating the different copies and for defining what “appropriate” is in any given context). For this to be achievable resources must be described in a way that describes the service they are offering (i.e. latest revision, full text, published version, work-in-progress version, abstract etc) using fully standardised or mapped semantics
- The file format of a resource provides some information as to the nature of a digital variant but few repositories highlight the relevance of any technical differences to the user (i.e. loss of information, compatibility with different authoring tools, file size, accessibility etc)

5.3 Access and authorisation control

- Another aspect of “appropriate copy” definition may be supported by fine-grained authorisation control i.e. users would have “see”-permissions associated resources only if they have appropriate permissions. This would require the repository to present the resources together in the user interface e.g. through a web page that represents a collection, and to be able to distinguish those which any user (or class of user) could have access from those to which they could not.

5.4 DigitalEdition identification

- Many repositories offer nothing in this domain, there is simply a replace mechanism; the system does not allow the user to see a history of either the metadata or of an older version of a resource (the binary)
- Some repositories provide revision/ digital edition management but this information cannot be shared with users without a login to the system
- Some repositories allow the user to manually change publicly available metadata records (there is no standardisation as to how this is achieved i.e. it could just be a comments field).
- Systems like SVN, CVS and wikis (e.g. MediaWiki) expose a full revision history of DigitalEditions but this is not expressed as a harvestable/ cross-searchable metadata record.
- The need for edition identification is perhaps most necessary in the harvesting model. A simple solution proposed by ePrints is to tag the latest version with a Boolean flag that can be used by harvesters and search engines to filter records to only show the latest Digital Editions

5.5 Workflow

- Some repositories provide a very basic workflow, mostly surrounding the process of collation of metadata and license statements to formally publish a resource
- A configurable workflow tool is very much a requirement for the future in the repositories services domain. Some repositories such as Harvest Road Hive do report a configurable workflow tool (however, we could not investigate this tool within the scope of this project).
- In general the repository community needs to define the kinds of workflow that would serve learner, researcher, teacher and administrator needs. This is not a simple task, as to benefit from effective workflow disparate groups will need to agree on a set of collaborative processes. The JISC RepoMMan project is exploring this area.
- The recent formation of the JISC Deposit API working group is looking to address the need for users to import objects from a range of "clients" into hosted and managed online repository software. WebDAV is a potential candidate for such an API and is being investigated by the JISC ASK and Spire projects. The API should support full revision control but ideally be coupled with file locking and authentication.

6 Conclusions

The issue of version identification is not simply (indeed not primarily) one of unique identification of resources but rather of defining the relationship between resources.²⁶ While it is important that each of those resources should be uniquely referenceable, from a user standpoint the more significant questions to ask are

- In what way are these two things the same?
- In what way are these two things different (and to what extent does that difference matter to me)?

As things stand, with objects in repositories, these relationships may be broadly discernable by the comparison of arbitrary attributes of apparently similar resources. However, without human intervention, even sophisticated search engines can be seen to be failing in this respect and we see little reason to think that this will change dramatically in the near future.

In the absence of human mediated services, the only alternative for a user is to examine resources for themselves and then to draw whatever conclusions can usefully be drawn from that examination. This model may work adequately for discovery purposes at some level but will ultimately prove to be unscalable and does not provide any basis for services which can provide a high level of predictability for users (low predictability of results inevitably leading to reduced levels of trust). In some circumstances, ambiguous, misleading or unreliable metadata may be worse from a user point of view than no metadata at all.

In this context, it is particularly important to remember that metadata isn't simply about discovery – it is about all aspects of resource management. There is a serious risk with equating metadata with discovery processes because that suggests that improved search technology could “solve the problem” – which it quite clearly will not or at least not within any usefully definable timescale.

Without standards for identification and standards for version semantics, technology will not prove to be a *deus ex machina* for version identity any more than it is for any other identification problem. Where policies are rule driven, technology may be able to help us to *apply* those policies effectively. But neither technology nor technologists can define those policies for us. It cannot be overstressed that decisions on “functional granularity”, although they can and should be rule driven, are matters of policy. For optimal interoperability, policies of this kind have to be uniform throughout a community. If they cannot be uniform, they must at least be clearly published and mapped so that a user can understand what rules are being applied and how.

In considering answers to the question of version identity, there is clearly a distinction that needs to be drawn between multiple versions of “the same thing” held in different repositories (under different management control) and several versions of the same thing held in the same repository. Interoperability becomes an issue as soon as different repositories are involved – where the use of the same semantics (or mapped semantics) becomes critical to mutual understanding; different metadata schemes – where the same relationship may be called by different names, or different relationships called by the same name – cause tremendous problems. Simplistic solutions like Dublin Core, while they have a place, must be supported by prescriptive semantics if they are going to fulfil this kind of purpose.

²⁶ Note this can be seen simply as a generic definition of metadata. See the <indecs> definition: “an item of metadata is a relationship that someone claims to exist between two referents”.
www.indecs.org

The challenge of workflows which cross different systems is particularly well illustrated by “formal publication” work flows – where the initial work flow may be in an institutional repository or on an individual academic’s desk top and the subsequent work flow within a publisher’s publication system, where there is no continuity of identity beyond some shared attributes (which may change). The proper solution to this lies in shared identification systems, but these depend either on authors applying consistent standards of unique identity before their work is placed into the publishers workflow (and on the publisher to maintain that identity); or on the publisher’s identity being applied to Ancestor versions in remote repositories (which is, by its nature, a human activity – both time consuming and error prone).

We have not been able, within the constraints of a scoping project of this kind, to develop a fully mature set of requirements for version identification, nor to define where priorities for activity are greatest. However, we believe that the considerable number of short use cases which we have gathered together in one place will prove a useful resource for future researchers.

Similarly, we have not been able definitively to assess the technical capabilities of the technology being implemented in institutional repositories in the UK and internationally. Nevertheless, so far as we can tell from our research, the technology will support the implementation of identification policies; the barriers are essentially human not technical.

The amount of content in institutional repositories in the UK remains low, and the challenge of retrofitting interoperable version identification policies to those repositories remains a manageable one in the short term. However, this will not be the case for long in the event that the trend to the proliferation of IRs – and the explosion of deposit of resources into those IRs – is in line with everyone’s expectations.

If JISC is to act effectively in this area, in terms of establishing appropriate policy mechanisms for interoperable version identification in IRs, the window of opportunity is not a very large one. Once very substantial quantities of content have been deposited, the retrofitting of a solution will become extremely costly.

7 Recommendations

We make here a limited number of high-level recommendations, which we believe to be appropriate to a scoping project of this kind.

7.1 JISC

1. We believe that this report will prove a useful tool for beginning a dialogue about the issues of repository identification. We therefore recommend that JISC should disseminate this report (or a suitably redacted version of it) widely to university repository managers and policy-makers.
2. It appears to us, from the limited contact that we have had with repository managers that awareness of the issues surrounding version identification and interoperability is not very high. However, that may be an oversimplification, and we recommend that JISC should undertake a more detailed survey into development plans for university repositories and awareness of versioning issues.
3. We have developed a high-level view of the requirements for version identification within IRs, but we are aware that much of this is not well evidenced and we certainly have had no metrics with which to prioritise requirements except our own assumptions. We recommend that JISC should research definitive sets of version identification requirements from researchers, teachers, learners, information professionals and other stakeholders before finalising policies in this area. Specifically, this should be designed to gain a better understanding of the way people are working with documents, images, movies, learning objects and other resources today, including the tools that they are using and the social groups (communities of practice) in which they work. Before moving into standards development, it will be essential to understand how people are using their desk tops, LANs, Web pages, simple repositories (eg eprints), and complex repositories (eg Fedora, CVS, Wiki) that allow sophisticated handling of workflows, revision and version identification. An understanding of social interoperability is critical to the successful implementation of technical interoperability.
4. The importance of the use of consistent semantics to clear communication and to interoperability has been stressed throughout this report. We have put together an overview of some candidate semantics which might be used, but these are emphatically not proposed as "the solution", simply as an input to the development of solutions. We needed these semantics simply to be able to communicate unambiguously *within the project*. We recommend that, once JISC has completed a more rigorous requirements exercise, that a more robust version of the various tentative taxonomies proposed in this report should be developed and implemented as widely as possible. This will require work with experts on as broad a basis as possible if interoperability is to be achieved between different stakeholder communities both nationally and internationally. This includes, for example, maintaining links with the NISO/ALPSP working group and contributing actively to the debate which will undoubtedly follow the publication of its report.
5. Once these various strands of work have been completed, it will be possible for JISC to develop framework policies which institutions can adapt to their own needs and to meet the requirements of interoperability. These framework policies should be appropriate for all content types, not just articles, theses and book chapters (in other words, policies must look beyond the workflow model of version identification).

6. We have identified one particular line of research – tools to support repository workflows and the version identification issues which relate to those – which might benefit from additional attention. We recommend that JISC consider extending work in this area.
7. Another considerable area of uncertainty relates to gaining a better understanding of the potential that search services, federated search and resolver technologies can provide to support users in identifying and locating the “appropriate copy”; this could also be a fruitful area for future research.

7.2 Universities (Institutional Repository managers)

1. We recommend that IR managers should consider the implications of introducing version identity management policies as early as possible in the development of their IRs, through careful consideration of the use scenarios presented in this report. Once JISC has created the framework recommended in (5) above, each institution will need to develop and communicate a clear policy on version identification, based on the framework. We recognise that version identity policies are simply one of a number of policies which will need to be developed (including, for example, policies on the persistence of resources deposited in IRs).
2. We recommend that each IR should develop and implement guidance statements for repository users (both depositing users and readers/researchers); and should provide the greatest possible support to depositors in following the guidance given (either through the use of very simple but intuitive user interfaces, and/or through expert human mediation). Again, we recognise that this is simply another facet of the problem of IRs and creation of standardised metadata.
3. It is clear that some uses of IRs would benefit considerably from instituting some form of workflow-based version management during the creation process; we recommend that IR managers should pay particular attention to this in specifying their requirements for IR development (see also JISC recommendation (6) above).
4. We recommend that IR managers should plan for search engine-driven access, where results may direct users to individual documents that do not have embedded version information. A solution to this would be to consider preventing direct access to documents, but providing access via a staging page carrying unambiguous version metadata.

Appendix 1: Sources consulted

Many of the following references were consulted as part of the initial desktop research, the results of which may be viewed at http://ask.oucs.ox.ac.uk/ask/index.php/Version_control.

JISC-funded Projects

ASK: Accessing and Storing Knowledge Project, <http://ask.oucs.ox.ac.uk/>.
CLADDIER: Citation, Location, and Deposition in Discipline and Institutional Repositories, <http://claddier.badc.ac.uk/>.
ePrints-UK Project, <http://www.rdn.ac.uk/projects/eprints-uk/>.
GRADE: Scoping a Geospatial Repository for Academic Deposit and Extraction, <http://edina.ac.uk/projects/grade/>.
MIDESS: Management of Images in a Distributed Environment with Shared Services, <http://www.leeds.ac.uk/library/midess/>.
Paradigm: Personal Archives Accessible in Digital Media Project, <http://www.paradigm.ac.uk/>.
PRESERV Project, <http://preserv.eprints.org/>.
Prowe: Personal Repositories Online Wiki Environment, <http://www.prowe.ac.uk/>.
RepoMMan project, <http://www.hull.ac.uk/esig/repomman/>.
Repository Bridge: Automated Linkage of National and Institutional Repositories, <http://www.inf.aber.ac.uk/bridge/>.
R4L: repository for the laboratory, <http://r4l.eprints.org/>.
SHERPA DP: Creating A Persistent Preservation Environment For Institutional Repositories, <http://www.ahds.ac.uk/about/projects/sherpa-dp/>.
SPECTRa: Submission, Preservation and Exposure of Chemistry Teaching and Research Data, <http://www.lib.cam.ac.uk/spectra/>.
SPIRE Project, <http://spire.conted.ox.ac.uk/cgi-bin/trac.cgi>.
STD-DOI: Publication and Citation of Scientific Primary Data, <http://www.std-doi.de/>.
StORe: Source-to-Output Repositories, <http://jiscstore.jot.com/WikiHome>.
VERSIONS: Versions of Eprints: user Requirements Study and Investigation of the Need for Standards, <http://www.lse.ac.uk/versions/>.

References

The Directory of Open Access Repositories – OpenDOAR, <http://www.opendoar.org/>.
Experimental OAI Registry at UIUC. Grainger Engineering Library Information Center at University of Illinois at Urbana-Champaign (2006), <http://gita.grainger.uiuc.edu/registry/>.
 "Identifiers" in *JISC Digital Repository Wiki*. UKOLN/JISC (2006), <http://www.ukoln.ac.uk/repositories/digirep/index/Identifiers>.
Interoperability Focus. UKOLN (2006), <http://www.ukoln.ac.uk/interop-focus/>.
NISO/ALPSP Working Group on Versions of Journal Articles, http://www.niso.org/committees/Journal_versioning/JournalVer_comm.html.
 "Scenarios and use cases" in *JISC Digital Repository Wiki*. UKOLN/JISC (2006), http://www.ukoln.ac.uk/repositories/digirep/index/Scenarios_and_use_cases.
A Technology Analysis of Repositories and Services: Project Repository. Johns Hopkins University, 2006: <https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository>.
<http://dublincore.org/documents/dc-citation-guidelines/>.
 Carpenter, Leona. "Repositories in Context: Digital repositories as components of an integrated infrastructure for education" (2005), <http://www.ukoln.ac.uk/events/delos-rep-workshop/presentations/carpenter.ppt>.
 Crow, Raym. "The Case for Institutional Repositories: A SPARC Position Paper". SPARC, 2002: <http://www.arl.org/sparc/IR/ir.html>.
 Deutsche Initiative für Netzwerkinformation e. V. "Dokumentenserver". (2005), <http://www.dini.de/dini/wisspub/dokuserver.php>.
 Dublin Core Metadata Initiative. DCMI Citation Working Group, <http://dublincore.org/groups/citation/>.
 Gibbons, Susan. "Current Landscape of Institutional Repositories". (2005),

- <http://www.aahsl.org/document/aahsl.ppt>.
- Green, Richard. "Iterative Development of Fedora 2.1 Materials (Draft)" RepoMMan Project, 2006: <http://www.hull.ac.uk/esig/repomman/downloads/D-D4-iterative-dev-0602>.
- Heery, Rachel and Sheila Anderson. *Digital Repositories Review*. 2005. UKOLN/AHDS (2005), http://www.jisc.ac.uk/uploaded_documents/digital-repositories-review-2005.pdf.
- Lagoze, Carl, Sandy Payette, Edwin Shin, and Chris Wilper, "Fedora: An Architecture for Complex Objects and their Relationships". Draft of submission to *Journal of Digital Libraries* Special Issue on Complex Objects (2005), <http://arxiv.org/abs/cs.DL/0501012>.
- Lynch, Clifford A. and Joan K. Lippincott. "Institutional Repository Deployment in the United States as of Early 2005". *D-Lib Magazine* 11:9 (September 2005): <http://www.dlib.org/dlib/september05/lynch/09lynch.html>.
- Miller, Paul. "Interoperability: What is it and Why should I want it?" *Ariadne* 24 (2000): <http://www.ariadne.ac.uk/issue24/interoperability/intro.html>.
- Mimno, David and Gregory Crane. "Hierarchical catalog records : implementing a FRBR catalog". *D-Lib Magazine* 11:10 (2005): <http://www.dlib.org/dlib/october05/crane/10crane.html>.
- Moreau, Luc, Liming Chen, Paul Groth, John Ibbotson et al. "Logical Architecture Strawman for Provenance Systems". Technical Report, ECS, University of Southampton, 2005: <http://eprints.ecs.soton.ac.uk/10796/>.
- Morris, Sally. "'Version control' of journal articles". NISO/ALPSP Working Group on Versions of Journal Articles, 2005: http://www.niso.org/committees/Journal_versioning/Morris.pdf.
- NISO. "Featured Working Group: Versions of Journal Articles". (2005), http://www.niso.org/committees/Journal_versioning/JournalVer_story.html.
- NASA/Science Office of Standards and Technology. "ISO Archiving Standards: Overview" (2006), <http://ssdoo.gsfc.nasa.gov/nost/isoas/>.
- Powell, Andy. "RDN/LTSN LOM application profile (RLLMAP)". Resource Discovery Network, 2005: <http://www.rdn.ac.uk/publications/rdn-ltsn/ap/>.
- Robertson, John. "Metadata and Digital Repositories: implementations -- repository systems". CETIS, 2005: <http://metadata.cetis.ac.uk/implementations/repositorySystems>.
- SPARC. "Select List of Institutional Repositories" (2006), <http://www.arl.org/sparc/repos/ir.html>.
- Van de Sompel, Herbert, Sandy Payette, John Erickson et al. "Rethinking Scholarly Communication: Building the System that Scholars Deserve". *D-Lib Magazine* 10:9 (2004): <http://www.dlib.org/dlib/september04/vandesompel/09vandesompel.html>.
- van Westrienen, Gerard. "Completed Questionnaires: country update on academic institutional repositories". Making the strategic case for institutional repositories: CNI-JISC-SURF Conference; Amsterdam, 10-11 May 2005, <http://www.surf.nl/download/country-update2005.pdf>.
- and Clifford A. Lynch, "Academic Institutional Repositories: Deployment Status in 13 Nations as of Mid 2005". *D-Lib Magazine* 11:9 (2005): <http://www.dlib.org/dlib/september05/westrienen/09westrienen.html>.

Appendix 2: Participants at workshop held 15 Feb 2006

- Ann Apps, MIMAS, The University of Manchester
- Mark Bide, Rightscom
- Peter Burnhill, Director of EDINA and Head of the Edinburgh University Data Library
- Les Carr, University of Southampton
- Michael Fraser, Research Technologies Service, Oxford University Computing Services
- Richard Green, Manager, RepoMMan Project, University of Hull
- Deborah Kahn, Rightscom
- Hugh Look, Rightscom
- Matthew Mascord, Project Manager, IBVRE: Integrative Biology Virtual Research Environment, Oxford
- Cliff Morgan, John Wiley & Son/ chair of NISO/ALPSP *Working Group on Versions of Journal Articles*
- Howard Noble, Educational Interoperability Specialist, Oxford University, Learning Technologies Group (LTG)
- Frances Shipsey, VERSIONS Project Manager, London School of Economics & Political Science
- Dave Price, Head of the Systems and Electronic Resources Service, Oxford University Library Services