Digital Library Context

L1. ONLINE APPENDIX TO PAPER PUBLICATION

A book/journal article is printed, with an online appendix (data set, textual apparatus, audio/video material). The printed material needs a persistent resolvable identifier for the online appendix.

- Book is printed. Contains printed link: e.g. "For Textual Apparatus, see hdl:123/321"
- Five years on, reader reads book, and types URI into an appropriate resolver service,
- Browser resolves to the online appendix
  - The resolver service appropriate for the identifier needs to be persistent, and discoverable. This may be done by including the persistent resolver in the printed representation of the identifier.
  - This use scenario assumes that users type identifiers, transferring identifiers from a non-digital to a digital medium. This is a non-functional attribute of identifiers Andy Powell identifies (Discussion Brief 2006): "Usable In Non-Digital Environments". Digital use of identifiers can avoid the need for users to type identifiers through encapsulation (users only ever follow hyperlinks embedding identifiers); but this is not always practical.
  - Semantically meaningful identifiers are easier for users to transfer from a non-digital to a digital medium. They are however problematic for persistence, as identified in the PILIN project.

L2. LIVE EMBEDDED METADATA

A digital object often needs to contain (or at least link to) metadata about itself: it is self-documenting as a self-contained object. If the metadata is changeable, linking to or embedding metadata through a persistent identifier allows the user to access up-to-date versions of the metadata. (TSO DOI report use case #2 & #3)

- A digital object is created. On its frontispiece, it contains inter alia a list of certifying authorities which have approved it.
- This list is changeable; it should not be frozen into the document as part of its content proper, but treated as metadata. It is updated by an authority separate to that updating the document content.

- The list of certifying authorities is itself an aggregate digital object, which can be formed from identifiers for the certifiers, plus metadata of its own (ordering, presentation).
- The aggregate object can be assigned its own identifier, which the content object calls out to, either through hyperlink or embedding.
- Updating the metadata is decoupled from updating the content.

L3. HYPERLINKED CITATIONS

A journal article in online (or paper) form has its journal references hyperlink to persistent resolvable identifiers.

- Journal article appears in online repository. Bibliography contains hyperlinks, e.g. "Nicholas, N. 2008. PILIN and the art of motorcycle maintenance. D-Lib 67:9. [ID]"
- Two years on, reader on browser clicks on the "[ID]" hyperlink
- Browser resolves to a gateway page for the Nicholas paper
- Gateway page may require authentication, or list different ways of accessing page, rather than presenting page itself (delayed accessioning)
- Important: citations are identifiers of FRBR manifestations, expressions, or works, and not items. The item the identifier resolves to may not be the most appropriate copy for a particular purpose, unless an explicit appropriate copy service has been deployed. Similarly, providing hyperlinks to items instead of manifestations or works is guaranteed not provide access to the most appropriate copy in the general case: I could be linking to a one-off Monash copy of an Elsevier work and satisfying the requirement of accessioning, but not the requirement of unique identification. Relevant to RQF use cases.

L4. DISCOVER FRBR-RELATED ITEMS

FRBR sets up an ontology of bibliographic constructs: Work, Expression (including Redaction), Manifestation, Item. Each of these can have their own identifiers, and the identifiers can be related through a FRBR relationship service.

- FRBR Items are identified and assigned locators. (Optionally, they could be assigned identifiers, but there is usually no compelling reason to do so.)
- FRBR Manifestations are abstracted from the copies and assigned identifiers. E.g. ISBNs are assigned to manifestations.

- FRBR Expressions and Works are abstracted from the Manifestations and assigned identifiers.
- Each of these higher-order constructs can be actioned to retrieve an Appropriate Copy of the FRBR Item.
- It is useful not to resolve a Work identifier to an Item by default, in order to educate users and to forestall their confusion between Work and Item. Best practice is instead to resolve work identifiers to bibliographic records, which then allows access to appropriate copies of Items.
- A FRBR service (such as the NLA is trialling: http://www.frbr.org/2006/10/20/nla) operates on the FRBR identifiers and the Item locators to list all the expressions and manifestations (and their associated appropriate copies) for a given Work.
- A FRBR service may also allow the retrieval of metadata attributes specific to each FRBR level, as set out in the FRBR standard. E.g. in music: works have keys, opus numbers, and historical contexts. Expressions have musical notations, reviews, and particular performers and recordings. Manifestations have playing speeds and recording technologies such as Dolby. Items have previous owners, physical conditions, and access restrictions.
- Such metadata allows e.g. Paskin's Use Case #8: "Discover and download version of French movie with French language sound track but English subtitles". This can be extended, through shared use of the identifier: e.g. "the movie and the subtitles must be from different providers".
- This is one kind of object relation expressible through a relationship service. E-Learning drafts, aggregations and customisations are another; e-research dataset–paper and paper–response relations are a third.

Note

The FRBR standard, section 5 details the possible bibliographic relations between FRBR items, and should be consulted for a relationship service to be used in a FRBR context. Note that works can be about works (literary criticism). Work–to–work relations can be: *successors, supplements, complements, summarisations, adaptations, transformations, imitations*. Expression–to–expression relations can be the same as work–to–work relations, as well as: *abridgements, revisions, translations, arrangements (music).* Manifestation–to–Manifestation relations can be *reproductions* or *alternates*. Item–to–Item relations can be *reconfigurations* or *reproductions*.

The FRBR standard section 7 should also be consulted: it is a functional requirements analysis for FRBR-based bibliographic systems on what the end-users will want to retrieve.

Note, as Paskin highlights: all metadata involve "a relationship which *somebody claims* to exist between two referents." FRBR relations are not always uncontroversial, so the authority of who decides the relation should be queryable, and multiple hierarchies should be possible.

In a fully FRBR-ised identifier system, such as described in Rehak, *The Appropriate Copy Problem* (http://hdl.handle.net/2000.01/D6E3BF9462684182AC293D64D3DDE192), resolution is defined as returning identifiers to metadata for the object, and identifiers to objects one FRBR level down: works resolve to expressions, expressions to manifestations, manifestations to instances, instances to locators. Practically, choices should only be presented to end users when there is a one-to-many relation. In Rehak's algorithm, the Select operator has enough selection criteria that at each level it can determine the single most appropriate path to follow.

Rehak suggests exploiting the one-to-many resolution of Handles as an implementation of the FRBR one-to-many mapping. The ontology Handle allows may not be sophisticated enough for all contexts, especially given limitations in current Handle proxy servers. Rehak allows the alternatives of non-actionable metadata to represent the relations, or a multiple resolution system external to Handle.

L5. RESOLUTION TO MULTIPLE MANIFESTATIONS.

A digital work can have several realisations of its content in different formats and presentations—according to user platform, user preference, disability access, etc. These are different FRBR manifestations of the same expression. The work or expression identifier serves to group the manifestation together. Individual manifestations can be accessed through parameterised resolution services, or through individual identifiers. If an actionable identifier provides multiple resolution, the appropriate manifestation can be selected through parameterised resolution, or resolution driven by user preferences. (TSO DOI report use case #5)

- A work—today's lecture on hippos—is published in three manifestations: a 200x400 pixel version, a 400x800 pixel version, and a text transcript. There is an identifier for the work, as an abstraction.
- The identifier for the work is capable of multiple resolution.
- The resolution service is configurable. Given user preferences on visual accessibility, a non-visual resolution (the transcription) is selected as the default resolution.
- A parameterised call of the resolution service on the work identifier can select a particular manifestation, based on discriminatory metadata (in an agreed scheme), rather than a distinct identifier; e.g. http://www.example.com/resolve/1.2.3/32893?res=lowres

- Alternatively, the resolution of the work identifier is to a selection, screen allowing the appropriate manifestation to be selected.

Note

It is possible that manifestations presented as alternatives for delivery differ not only in manifestation, but also in expression: i.e. that the alternatives differ in content as well as presentation. This can occur if alternate delivery requirements entail some abridgement or expansion of content. This can also occur if manifestations reflecting the latest version of content are not available for all deliverable formats ("we haven't converted the LOM record to Dublin Core yet"). The latter case is a contingency, and violates the functional requirement that any discrepancies in the content delivered should be motivated by a defined user requirement, and expected by the user.

Individual manifestations can have identifiers of their own. This only makes business sense if the manifestations will plausibly be accessed separately, rather than through a selector service as manifestations of the expression

L6. PATCH UP IDENTIFIER

A meaningful identifier scheme is employed: for instance, the Elsevier-pioneered Publisher Item Identifier (PII) (http://www.icsti.org/forum/23/index.html#identifiers) includes the ISSN or ISBN of the work, a year and issue number, and a check digit (e.g. S0165380696004038). The identifier lifecycle for a PII includes some checks, such as the prefix being S or B, the ISSN or ISBN number being registered, uniqueness in the namespace, and so on. But there is no check provided for someone mistakenly swaps two items PIIs. Someone does swap two PII. The identifiers are published before someone realises that the wrong issue got the wrong PII. Several pathways can be followed to address this problem, with increasingly negative side-effects.

- A meaningful identifier is created.
- Validity checks are run on the identifier.
- Nothwithstanding these checks, the identifier does not encode the correct meaning for its assigned referent.
    - Alt 1: The discrepancy is noticed immediately on publication. The identifier managers correct the identifier and republish. The identifier manager has the enforcement scope required to unpublish the old identifier—removing it from circulation, formally deprecating it, and so on.
    - Alt 2: The discrepancy is noticed after some time; the correct identifier has not been assigned to any thing, and the incorrect identifier is not meant to have been assigned to any thing in the

managed context (there is not something else it should be identifying). The correct identifier is registered, associated with the referent, and becomes the canonical identifier. The incorrect identifier is aliased to the correct.

- Alt 3: (Swap scenario) The correct identifier has already been assigned (incorrectly!) to another thing, and/or the incorrect identifier should have been assigned (correctly!) to that thing. The identifier manager has enforcement scope clout within the domain of identifier users, which is controlled. The manager swaps the identifier associations, and forces all users of the identifiers to swap the identifier names.
- Alt 4; (Swap scenario) The identifiers have been mistakenly swapped, and the identifier manager does control over the domain of identifier users (e.g. the identifiers have been released to the general public). The identifiers are left as is; at best, the identifier manager publishes metadata pointing out the discrepancy between identifier meaning and referent. The identifier manager can publish a new identifier overriding the mistaken identifier without swapping anything. I in cases like the PII, this still breaks the predictable semantics of the identifier.

Note

An example from Unicode. Unicode's character names are immutable identifiers, for obvious reasons. Some character names are known to be wrong, as documented in Unicode Technical Note 27 (http://www.unicode.org/notes/tn27/). For instance the Lao letters FO TAM "low tone syllable *fo*" and FO SUNG "high tone syllable *fo*" have been swapped. The best Unicode can do is follow strategy Alt 4: maintain the semantically wrong names, but add as aliases the identifiers FO FON "*fo* as in 'rain'" and FO FAY "*fo* as in 'fire'". Unicode was able to use the alternate meaningful mnemonic naming scheme in place for Lao.

Unicode has also mistakenly swapped LO LOOT and LO LING, which were already mnemonics. It cannot use the Lao mnemonic scheme, since these identifier names are already Lao mnemonics. Unicode's patch has been to make up its own mnemonic names, based on pronunciation: Lo and Ro. The errors in Lao were only noticed 15 years after the identifiers were associated, so swapping the character names back (Alt 2) was never an option. Given how embedded Unicode is as IT infrastructure, neither was forcing a name change (Alt 3) an option. (http://babelstone.blogspot.com/2006/03/unicode-character-names-part-3-name-by.html)

Unicode has further sought to mitigate this problem by claiming that the real identifiers for their characters are the codepoints (the hex codes)—U+0E9D, U+0E9F, etc, and the semantically meaningful names are conveniences; but this has not prevented user confusion. The problems with patching an identifier are a major motivator for avoiding meaningiful

identifiers, even as secondary aliases (see Online Appendix To Paper Publication).

L7. AUTHORISATION PROFILE

Certain items in a repository have a common access policy. That policy may be built up of several rules, each applying to an overlapping set of items. Each policy rule defines its own collection of objects accessible through that policy. Enforcing policy means identifying which policy-defined collections the object belongs to. (Paskin Use Case #11)

- A repository is governed by a set of policy rules.
- Each rule applies to a set of objects, and thus defines its own collection of objects.
- Each such collection has an identifier, and a membership service telling the caller whether the given policy applies to that object.
- Whenever an object is to be accessed, its authorisation is determined through these policy-collection–specific membership services.
- This allows Access Control List-style authorisation.
- The collections are virtual, and defined by their services, rather than a service resolving to a well-formed digital object. This is one of many instances of identifiers not defined through resolution to a locator.

L8. OVERLAY JOURNAL

Academics increasingly create their own overlay journals without needing to go through a publisher for infrastructure. Individual papers are accessed through institutional repositories; the overlay journal aggregates them and subjects them to peer-review, and provides value-add services.

- Author deposits journal article preprint to Institutional Repository, with a global identifier.
- Author sends preprint identifier to overlay journal editor.
- Editor sends identifier to reviewers for peer review.
- Reviewers obtain the preprint through the identifier.
- Author gets feedback from authors and generates final version.
- Final version is deposited in Institutional Repository with distinct identifier.
- Overlay Journal links to final version through identifier.

- Paper counts as 'published' in the Overlay Journal; publishing in the journal refers to certification from the journal. It does not refer to providing the content, which is still done by the institutional repository.
- The overlay journal may contribute other value-add services attached to the identifier, e.g. reviews, classifications.
- Draft and final versions may be linked as versions [see UPDATED DRAFT—PRESERVE LINEAGE scenario], or as expressions of the same work in FRBR terms (i.e. different services, not different IDs).

L9. MANAGED VIRTUAL COLLECTION

A virtual collection of digital objects can be assembled out of existing objects on repositories, independently of repositor managers. The result is a managed collection that spans different repositories.

- Collection manager identifies digital object to include in their virtual collection. Identification includes obtaining global identifier.
- Collection manager aggregates objects through identifiers as a collection.
- Collection manager assembles enough metadata and infrastructure that the collection can be queried or at least browsed. The collection manager needs not provide infrastructure to managed or store the objects in the collection: that remains the responsibility of the source repositories.
- Request to resolve identifier discovered through virtual collection will redirect to source repository.

L10. AGGREGATE DIGITAL OBJECT

A set of digital objects (e.g. individual courses or chapters of a monograph) are aggregated into a single digital object (e.g. a curriculum or a book). The component digital objects have their own identifiers. The aggregation organises those identifiers.

- A set of digital objects is identified for aggregation.
- Each object has an identifier.
- The identifiers are optionally put into a sequence, a hierarchy, or some other graph. (Cf. OAI-ORE, which represents aggregations explicitly as graphs of component objects.)
- The aggregation is identified with its own identifier. The aggregation resolves to the component object identifiers, their sequencing as metadata, and their relation to the aggregation through an Is-Part relation.

- The objects can be heterogeneous, which affects issues such as authority. For instance, a course built up of learning units made available by different authors has the contributors' authority for each contribution. The only authority the aggregator contributes lies in the metadata of the aggregation. (This is no different to an editor of a collection of studies in academia.)
- Similarly, the access privileges for components may be inconsistent with the privileges for the aggregate, and need to be resolved through off-band negotiation. The access provided for an object on a given repository will not necessarily be changed should a third party link to that object in their syllabus.

L11. FOLKSONOMY

An end user creates a folksonomy of digital objects, where each tag defines its own virtual collection. Unlike other virtual collections, folksonomies are not formally managed.

- Folksonomiser identifies digital object to include in their virtual collection. The identifier is registered by the digital object manager and not the folksonomier.
- Folksonomiser aggregates objects through identifiers as a virtual collection.
- Folksonomiser does *not* assemble the metadata and infrastructure that a managed collection would put in place for querying the collection. The only discovery enabled through the folksonomy is browsing of all the objects assigned a given tag.
- Request to resolve identifier discovered through virtual collection redirects to source repository.

L12. UPDATED DRAFT—KILL ORIGINAL

A new draft or update of a digital object is generated. It replaces the original.

- A digital object is created and deposited.
- The digital object is assigned a persistent identifier.
- In the author's private space, the author changes the original object to a revised object.
- The revised object is deposited, replacing the original.
- Both objects have the same ID, which resolves from that point on to the new object.

- Metadata may preserve some record of the updating.
- The original object is not accessible, archived, or discoverable.

Note

The distinction between a version of the same work, and a new derivative work, is determined by the authority responsible for the work. If the two objects originate from the same institutional authority, they considered versions of the same work, even if the actual authors are distinct. ("Version 1 was written by Adam for Link Affiliates; Version 2 was updated by Blair, also for Link Affiliates".) If the two objects have different institutional authority, they are considered different works. ("Work 1 was written by Adam for Link Affiliates. Work 2 was adapted from Work 1 by Blair, for the Ministry of Education, New Zealand.")

If Adam assents to the claim that Blair's object supersedes the earlier object, then Adam's object can be retracted, and the metadata claim that Version 2 supersedes Version 1 passes without comment. If Adam disputes the claim that Blair's object supersedes his object, however, the claim needs to be validated (to rule out malice and metadata noise), and assigned to an authority in a relationship service. This need not mean that the claim is suppressed if it is found inaccurate: the claim is of interest to any user of the object, and may still be discoverable and subject to testing.

L13. UPDATED DRAFT—PRESERVE LINEAGE

A new draft or update of a digital object is generated. The lineage of the document is preserved. [Ref. Van de Sompel slide 15: "Lineage: means to unambiguously express the workflow relationship between a new Digital Object and the one(s) it builds on."]

- A digital object is created and deposited.
- The digital object is assigned a rigid persistent identifier. (That is, it is specific to an expression, and does not resolve to different expressions depending on which is the most recent.)
- In the author's private space, the author changes the original object to a revised object.
- The revised object is deposited with a distinct rigid identifier.
    - Optionally: an identifier is created to act as a fluid identifier, identifying the currently active version of the work that both digital objects are expressions of.
    - Optionally: the fluid, rather than the rigid identifier is published. (This is PILIN recommended behaviour for a work identifier.)
- Metadata may preserve some record of the updating.

- A versioning/lineage service connects the two objects (either: the old and new rigid identifiers with the fluid identifiers; or: if a fluid ID is not created, the old rigid as a previous version of the new rigid ID).
  - The lineage service may not be public.
  - The previous version IDs may not be public. (So the fluid ID is global, the rigid IDs local.)
  - The previous version IDs may be public, but not resolvable to public content items. (Example: the Thesaurus Linguae Graecae digital library has substituted the text with identifier 0579 004 with the text with identifier 0579 012, and posted that change on its website. 0579 004 is no longer available through the repository. The TLG has met the requirements of preserving lineage, even though it no longer publishes previous versions of the document.)

Note

In FRBR Relationships terms, this is an Expression–to–Expression relation of Revision.

L14. HARVESTING SET

OAI PMH allows any subset of harvestable objects to have a set identifier (http://www.openarchives.org/OAI/openarchivesprotocol.html#Set); membership of the set is queryable by a harvester. More generally: arbitrary vocabulary items in metadata can be indexed through identifiers, and those identifiers are defined through non-resolving services.

- A harvesting set is defined on a repository through some membership criteria.
- The set is assigned an identifier.
- The identifier is operated on by membership services ("is item a part of set b?"), rather than resolving to a first order digital object: it defines an aggregation, and does not resolve to any one object.
- Membership services are required by OAI PMH to support services. The target repository must be able to respond with membership information when a set identifier is parameterised to the service requests *ListIdentifiers*, *ListRecords* and *GetRecord* .
- The identifier is used in the OAI PMH context, through these services.
- This is an example of a service operating on an identifier that is not a resolution service. Moreover, the identifier does not need a resolution service to be useful. Resolution is nonetheless recommended for accountability, possibly to definitions rather than digital objects—cf. Handle's registration of Handle types as Handles.

L15. LAPSED DOCUMENT OUTSIDE REPOSITORY

Bob Ingria put a paper up on his personal web page in 1996, that Nick Nicholas cited in his doctoral thesis. Ingria never gets around to publishing paper in scholarly press. Ingria dies in 2003. Paper was never included in a digital repository. Paper disappears, and the only remaining copy is a printout in Nicholas' filing cabinet. The paper is unpublished, and the identifier is unresolvable.

- Item is discovered on Internet.
- Identifier is registered for item by a party not involved with managing the item.
- Item locator unexpectedly breaks.
- Identifier manager realises this, and cannot find new location.
- Identifier manager does not have access to a copy of last resort. (Maintaining archives of content is the responsibility of repository managers and not identifier managers. Recall also that the item is not discovered in a repository—so noone has assumed responsibility for archiving it.)
- Identifier manager updates identifier: it now resolves to a stub, with whatever descriptive metadata the ID manager has retained.
- End user fails to access item itself through the persistent identifier.
- End user is not faced with a broken URL, but with the metadata that has been retained on the object status. In some contexts, this is adequate.

Note

Ingria, R. 2005. Grammatical formatives in a generative lexical theory: The case of Modern Greek *kai*. *Journal of Greek Linguistics* **6**. 61-101.

Persistence is inapplicable unless the data is managed by an institution that can outlive an individual. Universities are institutions, and in a sense so is the Wikimedia Foundation. An individual is not an institution.

A persistent identifier management system does not resolve the issue of persistent content; and there is little prospect of the identifier association being persistent if the object it is associated with is unmanaged. Moreover, if an identifier manager has no corporate authority (or curatorial responsibility) over the object being identified, managing the identifier will be problematic.

L16. LAPSED DOCUMENT INSIDE REPOSITORY

A learning object in a repository is declared obsolete. The object is taken offline, and the identifier no longer resolves. Unlike the previous case, this takes place within a managed object life cycle; and there may still be restricted accessioning of the object (e.g. for auditing).

- Object is deposited in repository and assigned identifier. Object is managed.
- Within object's lifecycle, object manager decides that the object needs no longer be discoverable on repository.
- Identifier is redirected to stub.
- Object locator is purposefully broken.
- Because object and identifier are managed, authority metadata is preserved and remains actionable.
- End user fails to access object itself through the persistent identifier.
- End user is not faced with a broken URL; they are assured at least that the object was retired purposefully.
- End user can escalate query given the actionable authority metadata, and can make arrangements for access if object is in closed archive.

L17. MIGRATE REPOSITORY

A repository migrates its repository from ePrints to Fez. In the process all the item locators need to change. The identifiers do not. This is a predictable and managed process of redirection.

- An object is deposited in a repository. A global identifier is mapped to its local locator.
- The object will be transferred to a new repository.
- Object is ingested from old to new repository.
- Global identifier is redirected to new locator.
- Old repository is decommissioned.
- End user clicks on identifier and gets same resolution as before.
- Persistence was guaranteed only because the object was managed—the repository manager was an actor who could initiate the identifier redirection in time. With an unmanaged object, this is impossible.

L18. THUMBNAIL HARVEST

A national library provides a thumbnail index to picture collections it harvests. The national library acts as a referator to those picture collection repositories. The thumbnails it creates based on the content object images are metadata: they are created through services from the originals, as a derivative object (FRBR manifestations), but serve to index the original. The thumbnails are distinct digital objects (as all metadata is), and can have their own identifiers. The thumbnail identifiers need not be derived from the original work identifiers; but a service must be available to map from one to the other. (That relation is the metadata discovery service.)

- Monash publishes Gippsland photos on its repository, each with its own identifier.
- The NLA harvests the Gippsland photos, though the repository of record remains Monash.
- The NLA forms thumbnails of the photos. These are new digital objects. They get their own identifiers.
    - Alternative: Monash supplies the NLA with its own thumbnails, as an exposed datastream of the same digital object as the photo.
- The new NLA thumbnails are treated as metadata of the Monash pictures.
- The connection between thumbnail and picture is made as a metadata resolution service at the NLA.
- The NLA metadata display for the digital object incorporates the thumbnail, used for resource identification by the user. The full digital object it refers to at Monash has its own metadata display, which may not incorporate a thumbnail.
- The thumbnails can be either fixed objects, or created dynamically. That does not affect whether they get their own identifiers, since identifiers can resolve to dynamic service invocations.

L19. DISCOVER VIA REGISTRY

An object is stored in a repository, and metadata for discovery is harvested into a registry. The object can be discovered through the registry, and the registry provides accessioning to the repository via the global identifier. Discovery is not disrupted by MIGRATE REPOSITORY.

- An item is deposited in a repository, and has a global identifier.
- The item metadata is harvested into a registry, along with the item identifier: the repositories are federated.
- The item can be discovered through a query on the registry.

- The query returns the global identifier, which is accessioned through the repository.
- The repository is migrated, and its locators changed; the identifier is updated.
- The registry is unaffected by this change, and no changes within the registry system are necessary.

L20. CONTACT INFORMATION

A digital object identifies its manager for authority purposes. The identifier needs to be fluid (the person filling the role may change), actionable (I should be able to contact the manager to escalate a query), and extensible (there are different ways to contact a person, and different attributes to do so with, such as contact hours). The contact information is captured in an aggregate digital object, with a global identifier.

- An aggregate digital object is created for the contact information of the PILIN Business Analyst. It contains a name, a landline phone number, a mobile phone number, an email address, a snail mail address, a room number, and a Skype address. It also contains contact hours.
- An identifier is created for this aggregate. Individual fields are accessible by specifying field names; through some parameterised service, and possibly a parameterised schema.
- Individual fields can be updated or added without disrupting existing use of the contact identifier: the outside world accesses this information through the contact identifier and a field parameter.
- The contact identifier can be used wherever a person needs to be identified, provided the context of use places no data model requirements on its resolution (or the resolution happens to match the required profile).
- This is an instance of resolution which does not entail a one-to-one mapping of a URL to a discrete digital object.

Note

The service for accessing this information can be a full web service. One can have invocations like:

- http://www.arrow.edu.au/contact/1159.1/62453?email returns ninichol@lib.monash.edu.au
- http://www.arrow.edu.au/contact/1159.1/62453?skype returns opoudjis
- http://www.arrow.edu.au/contact/1159.1/62453?snailaddress returns <address> <inst>ARROW Project</inst> <bldg>Building 4</bldg> <city>Monash University</city> <state>VIC</state> <postcode>3800</postcode> </address>

- Service calls can be orchestrated: http://www.arrow.edu.au/contact/1159.1/62453?skype can be parameterised in turn into a skype frontend through the skype API.

L21. ATTACH METADATA TO ITEM

Metadata is a digital object with an identifier. The content item described by the metadata is also a digital object with an identifier. Attaching metadata to an item can be done simply as an exposed relation between two identifiers.

- An item is created and deposited with an identifier.
- Metadata is created and deposited with an identifier, following a given metadata scheme (which has its own identifier).
- Since the metadata is a digital object, it is assigned an identifier.
- The metadata record is associated to the item it refers to as a pair of identifiers.
- The association is discoverable by the repository: given a query on metadata, the associated item identifier is recoverable.

L22. VERSION METADATA OF ITEM—KILL ORIGINAL

Metadata can be updated.

- A metadata record is associated with an object and a metadata schema.
- A new metadata record is created for the same object and through the same metadata schema.
- The new record is ingested into the registry.
- The new record is identified as equivalent to the current metadata record as they contain the same object identifier and metadata schema identifier.
- The original metadata record is deleted, and replaced with the new version.
- The metadata record has the same identifier, so the item–metadata association is not disrupted.

L23. VERSION METADATA OF ITEM—PRESERVE LINEAGE

Metadata can be versioned.

- A metadata record is associated with an object and a metadata schema.
- The association is indirect, through an aggregate object: this object, with its own identifier, has a field redirecting to a current metadata version, and fields for all the prior versions.
- A new metadata record is created for the same object and through the same metadata schema.
- The new record is ingested into the metadata registry.
- The new record is identified as equivalent to the current metadata aggregate record through common item identifier and metadata schema identifier.
- The new record identifier is added as yet another field to the aggregate metadata object.
- The current version field of the aggregate object (which is its default resolution within the repository) is updated to the new record identifier.
- The previous metadata record is preserved.
- The item–metadata association is not disrupted, since metadata is only accessed through the aggregate metadata object.
- Different versions of the metadata may have different authorities (different subjective judgements), so need not be considered as outdating one another. The decision on which version is active if any is then itself subjective.
- Extending this use case: the distinct metadata records need not even reside in the same repository or form part of an aggregate, versioned digital object. They can be completely autonomous from each other—though they would still need to be interoperable (through a shared referent identifier and well-defined schemas), if necessary.

Note

Cf. Scott Wilson's presentation "Why are identifiers for tins of beans difficult?" (CETIS workshop on identifiers). The ontology of learning object relations is: Copy, derivation, redescription; aggregation, disaggregation. There are other relations elsewhere, for bibliography [humanities] and e-Research.

L24. RQF—BIBLIOMETRICS

The Research Quality Framework wishes to investigate citation impact. CiteCorp is creating a citation discovery system to satisfy this requirement. Varying presentation formats for a citation are recognised and mapped to the same work. The mapping happens through an

identifier. That identifier has quantitative queries carried out on it to establish citation impact.

- CiteCorp gathers bibliographies from articles. References appear in every citation format available, from Turabian to Chicago to American Medical Association. These are dumped into a data bank as raw text.
- CiteCorp works out that a Turabian reference in paper 4, a Chicago reference in paper 7, and an AMA reference in paper 11, all refer to the same work.
    - Note that reference metrics are about FRBR works, not manifestations: it does not matter which edition my work was cited as, as long as it was cited.
- That work is assigned an identifier.
- The work is authored by one or more authors, who may also appear in the reference in different formats: full names, initials, etc.
- Each distinct author is assigned an identifier.
- CiteCorp offer a service mapping Author IDs to Work IDs to counts of references to the work.
    - CiteCorp's IDs for authors and works may well be at variance with my institution's, even if both are global. This constitutes a barrier to interoperability.
- Given the CiteCorp ID for my author (which may need some mapping), I can retrieve a citation impact score from CiteCorp.

L25. RQF—PANEL REVIEW: REFERATOR

The RQF will review submissions by panels with members from disparate institutions. Assume all reviewed content is open access. Monash builds not a repository of appropriate copies, but a referator to copies of content held in external repositories (i.e. an authoritative collection of links). In that context, the Work IDs, resolving to authoritative copies under rights management, are sufficient to enable access to the content by the reviewers.

- Monash identifies and gathers the best six papers over the past six years for Prof Shmuk.
- All Shmuk's papers are conveniently on an Open Access repository, each with an identifier globally resolvable to that content.
- Because the identifiers resolves openly, without access rights constraints, there is no need for a distinct locator for a local copy to be provided to reviewers.

- The identifiers are adequate for RQF purposes. Monash in effect ends up providing an Overlay Journal of Shmuk's content (or alternatively a portfolio) to the reviewers.
- The same holds if the object is not open access, but is located within a federation of repositories that the RQF reviewers also all have access to, with full accessioning. This is unrealistic if the reviewers are international.

L26. RQF—PANEL REVIEW: LOCAL COPIES

The RQF will review submissions by panels with members from disparate institutions. Profs Durak in Moscow and Fou in Paris will review 6 papers by Prof Shmuk in Monash. The submissions are identified persistently (over the lifetime of the RQF review). Monash builds its own repository with its own appropriate copies of the submissions, for the panel to review. The submissions may already reside in commercial closed repositories, e.g. Elsevier, with their own distinct identifiers. Monash does not have access to the Elsevier repository, let alone providing access to it to Durak. The appropriate copy must still be identified with the same global identifier.

- Monash identifies and gathers the best six papers over the past six years for Prof Shmuk.
- One of the papers is already in a closed-access Elsevier repository, with a global identifier m:n.
- Monash must provide Durak and Fou access to that paper. Monash cannot do so through make invididual piecemeal arrangements, and Elsevier is also disinclined to undertake such arrangements.
- Monash makes a local (in)appropriate copy of the m:n paper in its local repository, and exposes that copy to Durak, with the locator p:q.
- The identifier m:n is managed by Elsevier, and resolves to Elsevier. Neither Elsevier nor their ID manager are likely to cross-reference m:n to p:q.
- Durak wants to do his own bibliometric checking. He must have access to the m:n identifier to do so (even if he is not allowed to resolve that identifier to the Elsevier copy), because the identifier m:n is how the work is normally identified in the literature.
- Alternative 1: Monash exposes p:q as an identifier, and provides metadata to say that resource p:q is a FRBR item belonging to work m:n
- Alternative 2: Monash provides its own resolution service, in competition with Elsevier's, that maps m:n to p:q (plus metadata). This can take the form of an OpenURL service.

Note

Alternative 1 brings out several problems:

- I have to provide both m:n, a Work identifier, and p:q, an appropriate copy identifier.
- p:q identifies an item and not a work; items are typically not given persistent identifiers but only locators. However there is a requirement that p:q be persistent, which makes it an identifier.
- m:n is an identifier of a Work (or at least a Manifestation), not a Item; in this case m:n is being used not for its resolution to Items, but to track citations (which are of Manifestations). However end users usually require resolution of m:n to an Item. There is an endemic confusion between Items and non-concrete levels of abstraction by users, which extends to the citations users produce.
- Without validation, nothing prevents me from associating my own ID to a copyrighted item and calling it a new work. So persistent identifiers on their own don't solve rights management, although they can be used as a means of realising it (e.g. with strong validation techniques such as steganography).

The proper solution to the appropriate copy problem—getting an identifier to resolve the way a particular party prefers—is Alternative 2: Provide your own resolution service for a Work identifier, rather than a locator which foregoes the advantages of using the Work identifier.

Establishing a custom resolution service challenges the notion of authoritative identifier resolution, and requires a validation mechanism to confirm that the same objects are being identified.

Though Alternative 2 is preferable, it requires more coordination than Alternative 1, which is seen as a quick-and-dirty solution. The DOI JISC report (sect 1.2) acknowledges this kind of reality:

> Informal sharing of information resources is likely to have different digital identifier requirements from that of the more formal traditional publishing and dissemination processes. There is a need to provide, and assure the continued availability of, more informal methods of creating persistent digital identifiers, which have low cost and minimal barriers for information providers.

L27. APPROPRIATE COPIES—ONE SERVICE

An object is stored in two different locations. The identifier allows either location to be resolved to. The decision on which location to resolve to is up to the server, and can be informed by parameters to the resolution request.

- An object is deposited in a repository, and has a locator.
- The same object is deposited in a different repository, with a different locator.
- The two copies are kept in sync by the repository managers.
- An identifier is created which links to both objects through their locators.
- Resolution is provided by a service, which can pick either locator.
- The choice of locator that the service makes can be informed by repository uptime and physical location; digital rights; accessibility constraints; user preference; etc.

Note

See the RQF—PANEL REVIEW: LOCAL COPIES scenario. This scenario generalises that scenario's Alternative 2. Note that in the general case (as seen there) copies may be unauthorised, or at least not subject to the same authority.

The Grid term for appropriate copy is "replica", and the Grid would have a Replica Catalogue genre, enumerating all available replicas of a resource. Appropriate copy delivery is thus built in to the Grid's Obtain service.

L28. APPROPRIATE COPIES—LOCALISED SERVICE

An object is stored in two different locations. Two resolution services are set up. One resolves to one copy of the object, the other to the other (per RQF PANEL REVIEW—LOCAL COPIES). Each service has its own infrastructure, administration, and authorisation protocols. The only thing tying the two resolutions together is the identifier.

- An item is deposited in a repository, and has a locator.
- The same item is deposited in a different repository, with a different locator.
- The two copies are kept in sync by the repository managers.
- An identifier is created.
- Each repository provides its own service to resolve to their own copy.
- Each service has its own context, authentication, authorisation, management, infrastructure, etc. etc.
- The same goes for the digital objects the services resolve to; it is a social contract between the repository owners that ensures both are resolutions to the same object, and they needn't be (one service could resolve to an abridgement of the object).

- Even if one service is privileged, as a default resolution, the other can be invoked independently.

Note

This is the usual deployment strategy for OpenURL as an appropriate copy service in institutions: each institution has its own installation of the same service with the same interface. This scenario allows the localised resolution service to be quite distinct from the authoritative service, however. This allows great flexibility in what a user can do with an identifier within a specific domain; but the disparity in resolution services can lead to questions about the localised service's authority to resolve things as it choose, and indeed whether the same thing is being identified in both services.

It can also generate confusion over which party has authority over which service. Note that the localised resolution service is not dependent on whether the service host has permission or authority over the thing identified. (See following scenario.)

L29. INAPPROPRIATE COPIES

The popular and highly illegal BitTorrent referatory thepiratebay.org is unsatisfied with the ambiguity of its informal use of titles as meaningful content identifiers. It elects to use a canonical identifier for content. It obtains the identifiers from the copyright holders for that content, even though the referatory enables use of the content that the copyright holders object to.

- Copyright Thief obtains inappropriate copy of digital object.
- Copyright Thief obtains canonical identifier of object as maintained and published by the rights-holder (or their agent), and resolving to an appropriate, rights-managed copy: say the IFPI's Global Release ID A1-2425G-ABC1234002-M.
- Copyright Thief provides a service parameterised on the canonical identifier, and resolving to their illegitimate copy (as a bit torrent): say http://thepiratebay.org/avastmemateys/ifpi/A1-2425G-ABC1234002-M
- Copyright Theft Customers can use the IFPI's legitimate discovery services to discover the ID for the Work, and then use that ID on thepiratebay to find the corresponding bit torrent.

Note

As long as the public has access to an identifier, little can be done to restrict what people use the identifier for. The identifier has to be publicly accessible for discovery to work: the asset may be rights-managed, but the identifier cannot be. If one starts policing who can use their identifier, they end up killing the value-add capacity of the identifier.

L30. TIME-INDEXED PERSISTENT CITATION

A dynamic resource, such as a changeable web page, is cited according to a specific version. For highly dynamic or loosely managed resources (e.g. a managed resource with daily changes, or a web page), discretely identified versions are not practical. Instead, it may be desirable to index the version of the resource according to the date of publication or access —as has already become the norm in citation of web pages.

A service to resolve such citations is parameterized on the identifier of the resource itself, and a date index. Such a service presupposes access to versions of the resource harvested and indexed by date, and a fairly frequent period of harvesting, to forestall date granularity problems. The service generalises the functionality available at www.archive.org , and takes on the functionality of an Archival Identifier (ARK): ARKs allow versioning information to be embedded in the identifier as an optional component, though there is no explicit provision for a date-tagged system.

A resolution service parameterised for time retrieves a given date's version of a given resource. The harvesting it uses must be informed of changes in resource location over its lifecycle. Say hdl:102/34 points to a resource stored in Monash on January 2007, and in Toowoomba on July 2007. The resolution service request http://pilin.org.au/hdl/102/34?date=20070129 should resolve to an instance harvested from Monash, and the request http://pilin.org.au/hdl/102/34?date=20070729 should resolve to an instance harvested from Toowoomba, without the requester noticing any difference in access.

- A repository ingests all discrete published versions of a resource as they are published (or at least over a regular interval of low granularity).
- The repository allows time-indexed retrieval of the instance whose date of retrieval is closest to the specified time index.
- A global identifier resolves to an item in the time-indexed repository.
- The service to resolve the global identifier takes a time index as a parameter, and resolves to the instance in the repository whose date matches the time index the closest.
- The item migrates to a different time-indexed repository for future changes. The existing archive on the original time-indexed repository remains intact.
- The resolution service is notified of this change, and records the date of migration in its item metadata.
- Time-indexed queries indexed to before the date of migration are directed to the original repository; queries indexed to the date of migration onwards are directed to the new repository.

L31. CONFIDENTIAL DOCUMENT—EXPOSED IDENTIFIER

Typically, a document or other asset may have restricted access, but its metadata is open access: this allows discovery even if the user does not have immediate access to the resource. The subject matter of the document may be sensitive enough, however, that metadata revealing the subject matter (content metadata) must itself be access-restricted.

For instance, a dissertation with an identifier contains politically sensitive material. We allow the outside world to know that the identifier is an identifier, that it points to a document, and even that that document is a thesis at the given university; but we do not openly provide content metadata such as author name, title, abstract, or locator (which may be semantically-rich). So some metadata linked to the identifier, which would normally be maintained on the identifier system, needs to be subject to the same authorisation regime as the content object itself.

The identifier system then has two choices: either subject its own metadata to authorisation (possibly through a federated identity scheme negotiated with the content repository); or decline to store the metadata values locally, and refer all metadata queries to the content repository, which is responsible for authorising access. The authorisation step needs to take place before access to the resource is attempted: it should not be triggered by the locator, but by the identifier. This is to prevent the locator being divulged openly.

- A global identifier is associated with an access-restricted resource on a local repository.
- The global identifier harvests from the local repository the metadata that it needs to transact discovery, according to its normal profile.
- The global identifier also has “burnt-in” metadata, specific to the identifier itself, such as date of registration.
- The burnt-in metadata remains exposed to the public.
- The harvested content and resource metadata allow confidential information about the item content to be inferred. It is therefore subject to the same authentication for access as the content item itself.
- Whatever the mechanisms the identifier system has for querying the metadata, this metadata is not openly exposed to end users.
- Any block on exposing metadata is triggered by the identifier (which blocks the metadata record it is keyed to), rather than at the local repository (by which time the user will have worked out at least some resource metadata of the object).

L32. CONFIDENTIAL DOCUMENT: DARK IDENTIFIER

In the most paranoid instances, even the burnt-in metadata of an identifier is confidential: not only must the outside world not know anything about the referent or its location, but it must not even know about the identifier. The public should not know, without authorization, who requested the identifier, when it was requested, or what type of thing it identifies. In fact, it must not know prima facie that the identifier is associated to a thing at all. This makes the identifier "dark"—i.e. undiscoverable.

- A global identifier is associated with an access-restricted resource on a local repository.
- The global identifier harvests from the local repository the metadata that it needs to transact discovery, according to its normal profile.
- The global identifier also has "burnt-in" metadata, specific to the identifier itself, such as date of registration.
- No metadata is accessible by the public. This includes burnt-in metadata, which in turn includes the referent type, the date of registration, the identifier manager, and indeed the identifier string itself.
- Accordingly, browse requests for identifiers maintained through the identifier management system must exclude the dark identifier.
- Unauthorised metadata queries and attempts to resolve the dark identifier must resolve misleadingly, as if the identifier is unassigned. (A consumer can only discover the misdirection if they attempt to mint the identifier in the same namespace themselves, which makes them a trusted user.)

Note

If the user is authorised to register an identifier in the same namespace, they can discover that the identifier has already been assigned; but identifier registration is typically sufficiently restricted that this is not a concern.

L33. DEDUPLICATION

A repository wishes to guarantee that it does not accidentally ingest an exact duplicate of an object already ingested elsewhere in the same repository, or another repository in the same federation. Ingestion normally assumes the object is new to the repository and assigns it a unique identifier, as a novel work or expression. Unacceptable confusion result from two identifiers for the same content in the same repository, as the two instances each have their own metadata built up around them.

To forestall this, once a digital object is ingested, it is branded with the identifier; and ingesting within the federation checks that the candidate object (or datastream) has not already been ingested. The branding cannot be restricted to the metadata record, as the same content item can be submitted for ingestion using two distinct metadata records. The branding cannot be specific to an instance of the digital object: a copy of the digital object may have been made prior to submission, and then submitted independently for ingestion. Branding must therefore be determined by intrinsic attributes of object content.

- A digital object is submitted for ingestion in a repository.
- An identifier is associated with the digital object.
- The discriminant attributes constituting the association data for the identifier are not restricted to the locator: a duplicate copy will have a separate locator from the original, so locators do not prevent duplication in identifiers. Instead they depend on intrinsic attributes of the digital object content. (This is of course not applicable to FRBR Items, since the point of this scenario is to prevent discrete Items being confused for discrete Works or Expressions.)
- The association data for the identifier needs to be unique for the object, but common among all instances of the object. (A document title for instance is not necessarily the best choice of association data).
- Another digital object is submitted for ingestion in the same repository.
- The discriminant attributes of all objects ingested in the repository (or repository federation) are queried, to establish whether the object is distinct from all objects already ingested in the repository. If so, ingestion proceeds.
- If not, ingestion requires human intervention. The repository manager may decide not to ingest the record, to raise a query with the submitter, or to ingest the object as a (conceptually different) expression of the same work (cf. Borges' Pierre Menard).

Note

One possible implementation of this is message digesting: see DIGITAL SIGNATURE use scenario. This would not prevent a different version of the same work being ingested as a completely novel work, so deduplication should be combined with comparison of metadata attributes (such as common author–title pairs), and the metadata record should be subject to validation.

© Copyright 2007 | Legal | Privacy | Powered by ICE

The PILIN project is funded by the Australian Commonwealth Department of Education, Science and Training, (DEST) under the Systemic Infrastructure Initiative (SII) as part of the Commonwealth Government's Backing Australia's Ability – An Innovation Action Plan for the Future (BAA) under the ARROW Project.