LRE, ILOX, LOM and LAD

Integrating the ICIG architecture with emerging standards for interoperable learning repositories

Crispin Weston
11 January 2011

Version 1.0

Contents

Table of figures

Introduction

The European Schoolnet Learning Resource Exchange (LRE) is a specification designed to allow digital learning resources to be transferred between different repositories across Europe. At the time of writing, the LRE's most recent metadata application profile[1], version 4.5.1, combines a profile of Learning Object Metadata (LOM) with IMS Information for Learning Object Exchange (ILOX).

The purpose of this paper is:

- to give a brief overview of the LRE application profile, explaining how IMS ILOX integrates with LOM;
- explain how it is proposed to ensure that the work done in the SALTIS Interoperable Content Implementation Group (ICIG), based partly on work being done in LETSI on Learning Activity Definition (LAD), will integrate with LRE and ILOX;
- highlight areas in which the LRE might modify its model in order to improve its accommodation of packaged content;
- address particular questions which have arisen in the ICIG, notably about the negotiation of access to learning objects and the management of versioning.

[1] The current profile of the LRE Metadata Application Profile is available at http://lre.eun.org/node/6.

Executive summary

There have recently been two separate strands of work in the field of content interoperability:

- design of interoperable repositories for distributing learning content;
- improving runtime interoperability between VLEs and learning content.

Although these two strands often share similar data structures (particularly LOM metadata), they may have very different requirements in the way that metadata is used. For repositories, LOM represents a way of classifying content for the purpose of indexation; for content which has been packaged for import into VLEs, it is a way of describing content which has already been discovered.

Although the SALTIS ICIG is primarily concerned with the second of these two situations, it recognises the importance of producing outputs which are compatible with current initiatives for content distribution. At the same time, this paper argues that the LRE needs to take more account of the requirements placed upon distribution systems by learning content which:

- supports flexible aggregation and remixing by teachers;
- needs to exchange data at runtime with learning management systems.

Consequences for the development of ICIG LOM profiles

This paper provides a rationale for the inclusion of the following in the LOM profiles being developed for use by the SALTIS ICIG.

- Wherever allowed, LOM metadata should be located on the cloud and included within a content package by reference.

- LOM <technical.location> should not be supported for learning objects and should be optional for manifests.

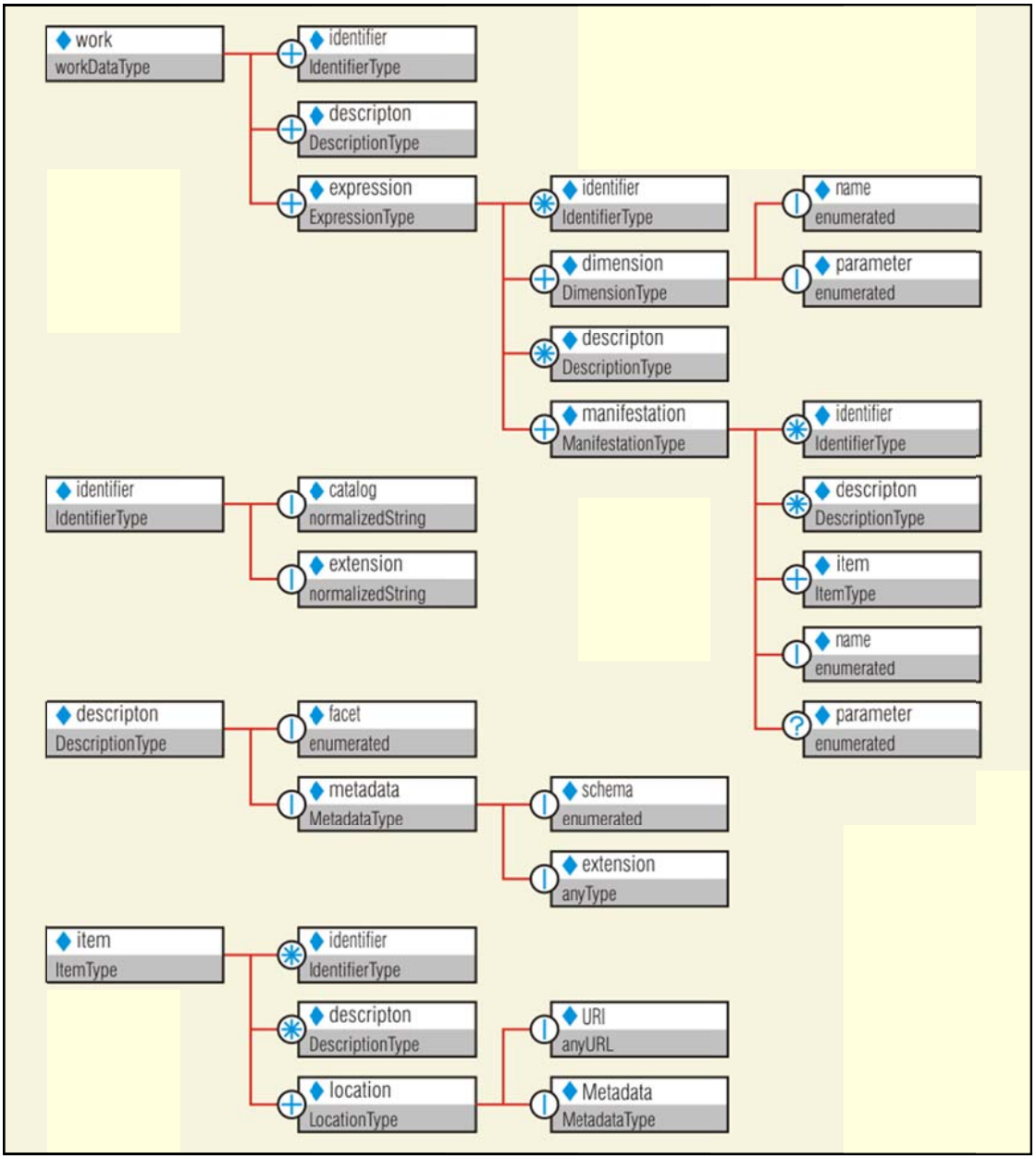
- A LOM <relation> of <kind> “isPartOfPackage” should be mandatory for packaged learning objects (i.e. requiring locally installed files).

- At least one LOM <relation> of <kind> “isListedInCatalog” should be mandatory for package manifests where that manifest is not made freely available through the use of a <technical.location> value.

- The ICIG should consider future use of a separate versioning object, probably proposing a new specification to replace ILOX. Such a versioning object could be referenced from appropriate LOM instances by a <relation> element of <kind> “isPartOfIloxWork” or “isVersionedBy”.

An overview of the LRE profile of IMS ILOX

The root of the ILOX hierarchy is the <work> (in the sense of a work of literature). Each <work> can have one or more <expression> records, each <expression> being distinguished from its siblings in terms of one or more dimensions—language, accessibility, version or coverage. Each <expression> can then have one or more <manifestation> records, each analogous to a format. These might be identified as Common Cartridge, IMS Content Packaging, “preview”, “thumbnail” or “printable”. Some of these values represent genuine alternative formats of the main work, while others (“preview”, “thumbnail” and “printable”) represent supplementary renderings. Each <manifestation> then has one or more <item> records, each <item> representing one particular physical location of the parent <manifestation>.



Each level of the ILOX tree can have a number of <description> objects attached, each <description> being distinguished from its siblings by a <facet> value, selected from a controlled vocabulary, which includes “main” and “license/rights”, amongst others. Each <description> wraps a metadata instance. In the LRE profile, the mandatory “main” <description> wraps a LOM instance.

License information

The root <work> node must have license information, carried either within the <rights> section of the LOM instance wrapped in the “main” <description>; or in Digital Rights Expression Language (DREL) wrapped in a separate “license/rights” <description>.

LOM is sufficient to represent license information where the resource is open or where it is covered by Creative Commons. In this latter case, the LRE LOM profile uses LOM element 6.3, <rights.description>. If this is stated to be of pseudo-language “x-t-cc.url”, then the associated string will be interpreted as a URL pointing at a Creative Commons license, for example “http://www.creativecommons.org/licenses/by-sa/it”.

If LOM <rights.copyrightAndOtherRestrictions> equals “yes” and there is no LOM <rights.description> of pseudo-language “x-t-cc-url” then the LRE rights information must be encoded in the ILOX “license/rights” <description>.

If license information is included at lower levels of the ILOX hierarchy, these more particular instances override the more general rights information at the root <work>.

Limitations of the LRE profile of ILOX

While there is a clear requirement for the version-management, it is not clear that ILOX is the right solution.

1 Coupled to LOM

Under the current LRE profile, no metadata is provided that supports learning objects requiring runtime support (see *An overview of Learning Activity Definition* on page 14).

2 LOM is embedded not referenced

Under the LRE profile, LOM metadata is embedded within the ILOX XML. This introduces a degree of inflexibility into the design, requiring the ILOX to be used as the entry point and preventing ILOX from providing a versioning service to metadata models with different entry points (such as from a content package).

3 Optional identifiers

Identifiers are optional at all levels of the hierarchy except the top level of <work>. This makes it difficult to link back from an external LOM instance to a corresponding <expression>, <manifestation> or <item> in the ILOX hierarchy.

4 Questionable requirement for some types of <expression>

While there is a clear requirement to manage the versioning of languages and updates, the inclusion of “accessibility” and “coverage” dimensions is questionable.

ILOX is suited to version management at a macro level, prior to the acquisition of a product. Differentiating between accessibility requirements will generally need to be managed at a micro, per student level. The ability to purchase an “accessible version” of a work may be useful in certain circumstances, it is unlikely to be helpful in the case of learning content for use in a general classroom. It is therefore not clear how the provision of alternative accessible versions of a <work> will it into an overall, coherent strategy for accessibility.

The inclusion of the “coverage” dimension is also questionable. This allows different <expression> structures for the same <work> where those different versions have different coverage. Most people would consider that different coverage is a characteristic of a different work: the Baedeker guide to France and the Baedeker guide to England are different works, not different expressions of the same work.

5 Non-hierarchical treatment of language and update versions

The two dimensions of <expression> that are likely to be useful are “language” and “version”. However, these are organised in a non-hierarchical way, so that a list of four different expressions might read:

Expressions	English language, version 1.0
	French language, version 1.0
	English language, version 2.0
	French language, version 2.0

Figure 2. Organisation of alternatives in a flat list

A more structured and scalable representation of the same versioning information would be as follows:

Language	English	Version	1.0
			2.0
	French	Version	1.0
			2.0

Figure 3. Organisation of alternatives as a multi-dimensional hierarchy

In any hierarchy designed to manage versioning, it is desirable in the interests of efficient traversal of the versioning hierarchy that each level of the hierarchy should only manage a single variable.

6 Unclear requirement for <manifestation> level

The <manifestation> level of the ILOX hierarchy is also used for multiple purposes:

- providing the main work in different formats;
- providing supplementary objects such as thumbnails, previews and landing pages (web pages providing information about the <expression>).

The requirement for supplementary objects is not a versioning requirement. The essence of a versioning service is a logical "or"; the essence of supplementary objects is a logical "and". This leads to two difficulties:

- suppliers may wish to provide thumbnails, previews and landing pages for learning objects which are not versioned and which do not require any ILOX records;
- publishers may wish to provide separate thumbnails, previews and landing pages for the different formats—a requirement which would not be met by the current proposal to treat these objects as further formats.

It therefore makes sense to provide links to supplementary objects from the LOM or LAD records and not from the ILOX.

The actual demand for multiple formats is also questionable. The overwhelming demand from publishers[2] is to consolidate packaging standards so that publishers are not faced with a proliferation of minor variants of packaging standards.

In addition to these particular objections, it should be noted that ILOX is based on the Functional Requirements for Bibliographic Records, which has had a poor record of implementation in repository projects since it was conceived in 1998. In 2009, Talat Chaudhri noted that "FRBR was designed for library catalogues, not repositories. The purposes and requirements of Web delivery of resources through repositories are very different to those of library systems"[3].

Different approaches to versioning

The topology of different kinds of versioning

There are two clear requirements for the management of versioning:

- language versions;
- update versions.

There are two significant differences between these two types of version:

---

[2] See *Interim Report on Content Publisher Interviews* at http://www.saltis.org/papers/papers_cp.html.

[3] "Assessing FRBR in Dublin Core Application Profiles" at http://www.ariadne.ac.uk/issue58/chaudhri/.

- of all update versions there is only one (the most recent) which is recommended for all users, while for language versions, different users will at any particular time prefer different versions;
- consequently, users' interest in update versions will vary as updates are produced, while their interest in language versions will generally remain constant—French learners will always prefer the French language version.

The fact that of all update versions, there is only one that is recommended makes the creation of a separate versioning object unnecessary: all versions can simple include a single pointer to the most recent update. However, in the case of language versions, the fact that there is no single recommended language version makes the use of a single pointer inappropriate.

If a separate versioning object is considered necessary from the point of view of managing language versions, it makes sense to use it for update versions as well. For this second reason, the ICIG should look to a versioning method which uses a separate versioning object. This could either be a revised profile of IMS ILOX or a fresh approach to the creation of a versioning object.

Recursive LOM

As part of the ASPECT and ICOPER programmes, Frans Van Assche, Joris Klerkx and Erik Duval have proposed what they call "recursive LOM"[4], where one LOM instance can be nested within another, the parent LOM containing those elements which are common to all versions, and nested instances containing those elements applicable at more particular levels of the hierarchy. This proposal extends LOM to provide its own native versioning control.

Assche, Klerkx and Duval propose two mechanisms for managing the nesting of LOM instances:

- using separate metadata instances which are linked through the use of a <relation> of <kind> "hasmetadatapart", the subordinate record possibly containing a reciprocal <relation> of <kind> ispartofmetadata";
- using embedded metadata instances, child instances being contained within a new section of the meta-metadata section called <meta-metadata.metadatarelationship.lom>.

The proponents of recursive LOM prefer the second solution on the grounds of "conceptual cleanness"—and it is only the second solution that is given a means of distinguishing the purpose of different subordinate records, using a new element <meta-metadata.metadatarelationship.relationshiptype>— an enumerated value which can contain terms such as "language", "format", or "accessibility". It is perhaps a disadvantage of this proposal that the entire subordinate record must be read in order to discover what particular language, format or accessibility options the subordinate

[4] "How to describe multiple versions of the same" at https://lirias.kuleuven.be/bitstream/123456789/280117/1/edmedia-same.pdf.

instance addresses. This would lead to the repeated loading and parsing of irrelevant LOM instances.

A more serious problem with the recursive proposal is the fact that the LOM instance is split up into several separate records. The authors optimistically rate the proposal as "excellent" in terms of "fitting in with current LOM practice". In fact, any uninitiated LOM user would fail to read all but one of the interdependent records, effectively and unknowingly reading an incomplete LOM instance. This might have serious consequences, leading legacy systems to import content without realising, for example, that it was subject to licensing conditions.

Objection to separate versioning objects

The objection made by Assche, Klerkx and Duval to ILOX is that it breaks compatibility with existing specifications, such as IMS Content Packaging, which expects to reference LOM instances. While Content Packaging can reference any kind of metadata, it is true that systems designed to implement Content Packaging may require further development if they were to handle ILOX references. As already discussed under*2LOM is embedded not* referenced on page 4, even this objection would only apply while the LOM instance is wrapped by the ILOX. The difficulty would be overcome by a separate versioning object which referenced a separate LOM instance, allowing a reciprocal reference from the LOM to the individual versioning node.

If at the same time the versioning record avoided any metadata roll-up, specifying instead that only complete LOM instances could be referenced from leaf nodes of the versioning hierarchy, then the problem noted above with backwards compatibility would be avoided.

Proposal for a Simple External Versioning Object

The following is provided as a straw-man proposal for a new Simple External Versioning Object (SEVO) which would avoid the major disadvantages of both ILOX and recursive LOM:

- it avoids inefficient, multi-dimensional flat lists;
- it references external LOM instances and therefore does not break existing LOM implementations;
- it avoids breaking LOM instances into constituent parts, which risks corruption of data when imported by legacy systems;
- it provides all information required to select an appropriate version within the versioning object, preventing the requirement to read of redundant LOM instances;
- it avoids the confusion in ILOX between the provision of alternative versions and supplementary artifacts.

SEVO is proposed as having three element types:

- the <work> is always and only at the root of the SEVO hierarchy;
- the <instance> always and only provides the leaf elements of the SEVO hierarchy;

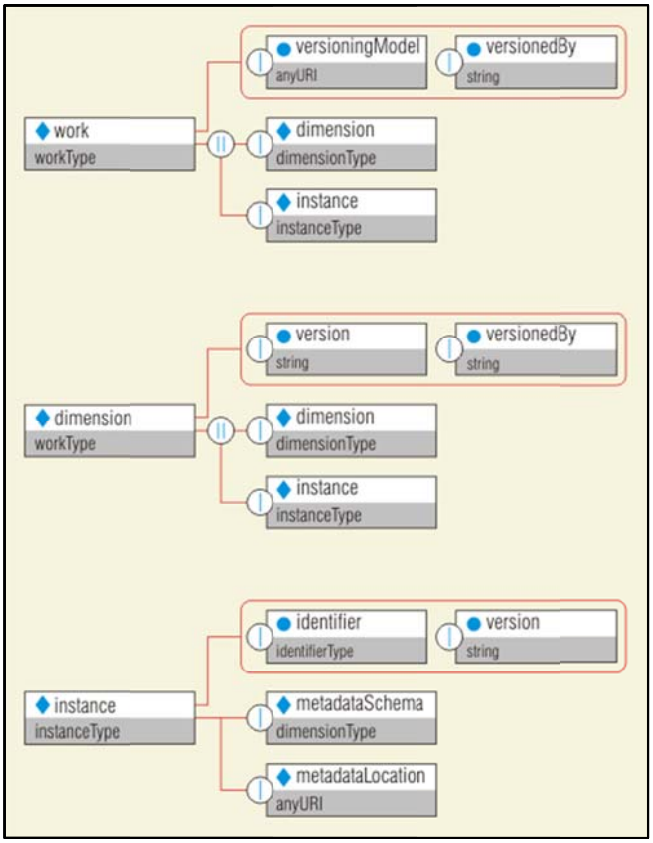
- the <dimension> provides the intermediate elements, where these are required by multi-dimensional versioning objects.

Each parent element (either a <work> or <dimension> has a @versionedBy attribute which can have an enumerated value such as “language” or “update”.

Each child element (either <dimension> or <instance>) has a corresponding @version attribute, valid values for which would depend on the value of the parental @versionedBy.

Each <instance> element has:

- an @identifier, enabling a reciprocal reference from any external metadata instance, along the lines of “http://www.xyz.com/sevo.xml#identifier”;
- a <metadataSchema>, representing the type of metadata which is being referenced;
- a <metadataLocation>, containing the URL of the referenced metadata itself.



The only obvious drawback of this proposal is the difficulty in ensuring valid values for <dimension>.@version and <instance>.@version. One way of addressing this issue would be to provide a second schema which might be called a “Simple External Versioning Model”. This would provide a list of acceptable terms for the @versionedBy attribute, nested within which would be information on acceptable values for associated @version attributes. A reference to a SEVM instance would then be included as an attribute to the <work> element at the root of any individual SEVO instance.

Omissions from the LRE model

Although the LRE/ILOX model provides a useful way of versioning content, it is blind in two critical respects:

- the aggregation of learning objects;
- different runtime capabilities of learning objects.

Aggregation of content

The weakness in addressing aggregation reflects the weakness in the way that IMS Content Packaging and its derivatives have developed in this same respect. The original conception of a "package" is that it should be a container for multiple objects. Current IMS advice, however, is that a Common Cartridge should contain only a single <organization>[5] and that if a package contains multiple <organizations>, that these should represent alternative "views" of what is fundamentally the same content[6].

A further restriction is that content packaging standards have evolved in a way which restricts the different types of content which may be included in the same package. This means that any product or course which consists of different types of content (e.g. QTI and SCOs) must necessarily be transported in different packages.

This retreat from the original conception of a package as a container is reflected in the LRE documentation, which represents different content packages as different "manifestations" or file formats of a single <work>, rather than different types of container able to contain multiple <work> objects. The issue of aggregation, and strategies for its management, is almost entirely absent from the LRE profile[7].

There may be up to five different functional levels associated with an aggregation hierarchy:

- the smallest unit of content that can be used independently (the learning object);
- the unit of content which is versioned (the ILOX work);
- the unit of content which is transported (the content package);
- the unit of content which is purchased (the product);
- the unit of content which covers a defined area of the curriculum (the course).

The LRE profile is that it assumes that all these functional points coincide in the same learning object, when they may in fact all occur at different points in the aggregation hierarchy.

---

[5] Section 4.1 of Common Cartridge (available from http://www.imsglobal.org/specifications.html) says that "A Common Cartridge may define a single organization or include no organizations. Multiple organizations are not permitted…".

[6] Section 6.5.2 of IMS Content Packaging version 1.2 (available from http://www.imsglobal.org/specifications.html) says that "Multiple Organization objects are equivalent in purpose. Each shows a way for structuring the same set of Resource objects within a given Manifest".

[7] Under their proposals for "access control", LRE learning objects can be given a "collection identifier". Although "collection" is used with widely differing meanings throughout the LRE document, in this case it represents a single access point (or "product"). There does not appear to be any single metadata record where information about that product is stored.

Types of learning object

The runtime integration of learning objects with a new generation of capable learning management systems is a critical requirement[8] for the delivery of effective digital learning in the context of formal education. While LOM metadata allows for the inclusion of information about the MIME format of a particular learning object, it is blind to the issue of runtime management: the need for Learning Management Systems (LMSs) to exchange contextual data with the learning object as it is run. Such runtime data may include:

- scores,
- state,
- information about the learner,
- information about co-participants in the learning experience,
- creative product,
- parameters which adapt the way that the object should run.

The different types of runtime data which are supported by a particular learning object, and which may be essential to its proper functioning, add a new dimension to the understanding of object "type", which needs to be supported by repository systems.

Need to recognise multiple purposes of metadata

The LRE/ILOX profile supposes that the only purpose of Learning Object Metadata is to be read by an indexing service. It consequently embeds LOM instances within ILOX structures, assuming that indexing services will approach the LOM instance through the ILOX wrapper. LRE is silent on the fact that the content packages which it references at the leaf <item> nodes also include LOM instances, which will, under the LRE profile, necessarily duplicate information in separate locations.

Package authors use LOM for a different purpose to managers of repositories. While repositories use LOM to allow users to discover content, content packages user LOM to describe learning objects to which the user already has access.

These two different use cases suggest:

- different priorities in the profiling of LOM;
- different ways of accessing a LOM instance: LOM instances will be accessed by LRE repositories through an ILOX wrapper, while they will be accessed by ingesting LMS systems from within a content package.

The first problem can be resolved, either by agreeing a common profile of LOM, or by distinguishing between LOM instances which are intended for use in a packaging context and LOM instances intended for use in an indexing context.

The second problem can be resolved by:

[8] See Interim Report on Content Publishers Interviews at. http://www.saltis.org/papers/papers_cp.html.

- extracting the LOM instance from the ILOX wrapper and placing it independently on the cloud;
- referencing the record from multiple points, which will include the ILOX versioning record and the content package.

The need for multiple, interlinked data records

The extraction of LOM metadata from the ILOX wrapper highlights the need for an overall architecture which recognises:

- the existence of multiple data records;
- the fact that these different data records are interlinked in a way which allow the user to navigate between them.

This architecture based on multiple interlinked data records is consistent with current trends towards the semantic web. A brief list of the different types of data record required to address some of the problems identified above are:

- LOM instances;
- versioning records such as ILOX or SEVO;
- manifests which conform to IMS CP and its derivatives;
- records managing levels of aggregation not supported by current content packaging specifications, such as LAD courses and catalogs;
- records providing flexible descriptions of new runtime behaviours.

Acquisition of indexed content

Repositories use LOM to create searchable indexes of content. These are of little use if the search results are not linked to some mechanism which provides access to the resource once it has been discovered. LOM provides a means of access through the <location> element, which gives the URL of the actual resource to which the LOM instance applies.

Problems with LOM <technical.location>

There are a number of difficulties with the LOM <technical.location> element.

- Where the resource is intended for distribution on disk, it would not be possible to give any globally-valid URL in the LOM <technical.location> element, unless the supplier of the learning object wanted to provide a default copy on the cloud.

- In the case of a commercial product, it is unlikely that the package author will want to provide direct access even to a default version of the resource itself. Where a single URL does exist, it is likely to be protected an authentication process, and will therefore be useless to most users of a searchable index. To cater for this case, the tagging guide for Curriculum Online specified that this

should be “a URL linking to where the resource *or information about it* can be found” (italics added)[9]. While recognising the need to reconcile OER and commercial suppliers, this formula introduces a degree of ambivalence in the interpretation of the <technical.location> field which undermines interoperability.

- In the case of commercial content, the object to which access may be granted is unlikely to be the same as the subject of the LOM instance. The user may find an individual learning object but in order to access this object may be required to purchase a larger product or subscription. In other circumstances, the user may discover an object which he may not realise is already contained within one of the user’s existing subscriptions.

- A similar difficulty is raised by a versioning service like ILOX, where the LOM instance corresponds to an ILOX <work> but where it is unclear which version or translation of that abstract <work> the user wants to access. In this case, the ILOX hierarchy would need to be traversed in order to locate the acquisition information for the appropriate version.

- A learning object may require integration with a Learning Management System, for example using the SCORM runtime. These are technical requirements which cannot be encoded within traditional LOM metadata and which may make the learning object unusable in the searcher’s current environment.

LRE transactions

The difficulty regarding the negotiation of access to commercial content is addressed by the ILOX-based LRE model, which allows the publisher to attach detailed access control information using a <description> with a <facet> of “transaction” at an appropriate level in the ILOX hierarchy.

It is a weakness of the LRE proposal that the transaction metadata is attached to an individual <work> rather than to a complete product.

SALTIS may wish to contribute to the future development of this specification. While the specification remains immature, a simple fall-back position would be to use URL links to “acquisition pages”—web pages which enable the negotiation of access to the specified object. These are similar to what the LRE calls “landing pages”—although the LRE profile does not specify that landing pages must allow the negotiation of access.

[9] See archived version at http://industry.becta.org.uk/display.cfm?resID=40282.

An overview of Learning Activity Definition (LAD)

The LETSI Learning Activity Definition (LAD) project is addressing issues both of aggregation and typing of different kinds of learning object.

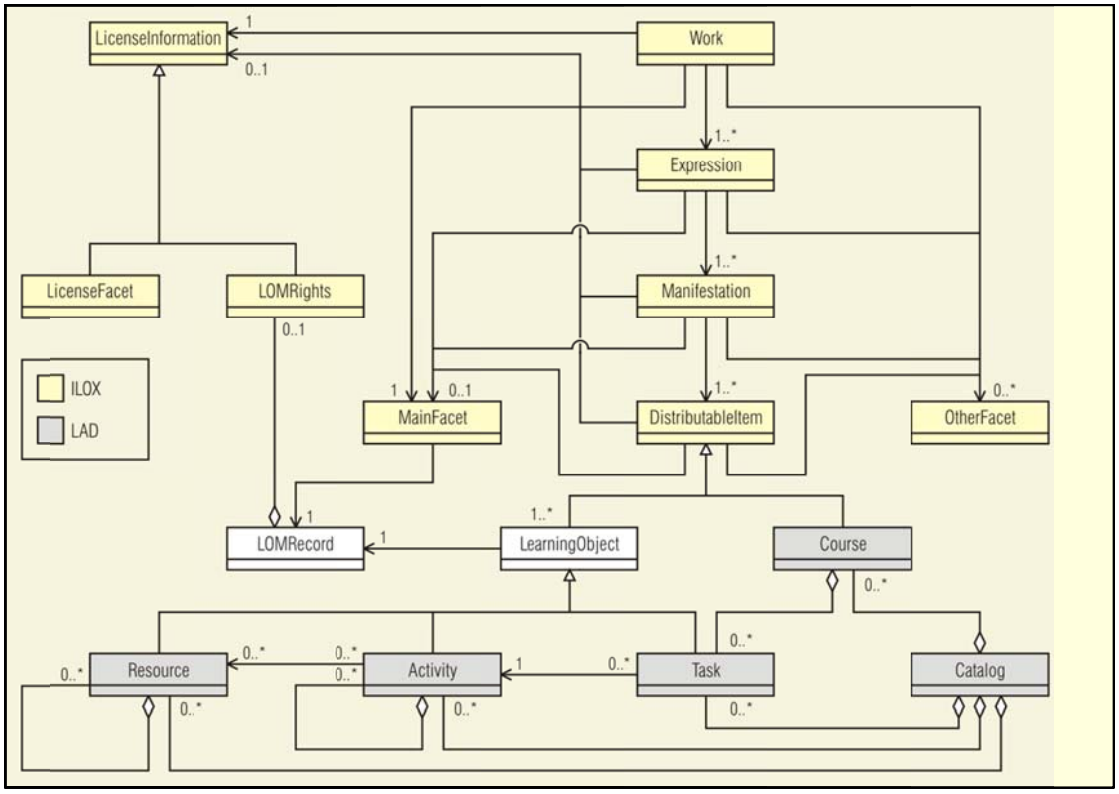


Figure 5 shows how a small section of the LETSI LAD ontology could be brought together with the LRE/ILOX scheme. To avoid ambiguity, the ILOX *item* has been renamed in this diagram the *DistributableItem*.

Different types of learning object

LAD proposes three fundamentally different types of learning object.

- Resources represent digital materials and services intended for use in a learning environment.

- Activities represent a digital capability to deliver a consistent learning experience to appropriate learners. Each activity may reference any number of resources. Where activities require inputs, default values must be provided to ensure viability of the activity. Activities may produce outcomes.

- Tasks represent the contextualisation of an activity within a particular formal learning environment. This may, for example, link the outcomes of the activity with a particular set of competency definitions, appropriate to the local learning environment. In a formal learning environment, tasks are what teachers assign to students and correspond to columns in a conventional grade book.

It should be noted that LOM implicitly acknowledges the distinction made by LAD between activities and tasks. LOM <general.description> is meant to describe the underlying learning object (i.e. the resource or activity) itself; while LOM <educational.description> is meant to describe the *use* of the learning object (i.e. the task). LAD argues that these different types of <description> should not be contained within the same object because a single activity might be used in a variety of different ways, many of which may not be foreseen by the original publisher.

It should also be noted that the series of learning object classes, from *resource* through *task*, represents a movement from informal to formal learning environments. The *task* represents a unit of learning content which is "baked" and ready for a specific kind of use by a non-technical teacher. By contrast a *resource* represents a program, service or unit of content which may be used in many different ways, providing a reusable object for informal learners who do not wish to be constricted by a particular set of instructional assumptions, or for the technical enthusiast who wishes to adapt content to new pedagogical purposes or methods.

Different types of aggregation

LAD proposes six types of aggregation.

- Resources may be combined to create composite resources. For example, pages and chapters may be combined to create a book.

- Where an activity references more than one resource, the resources may be said to have been combined, along with appropriate instructions, to create a single activity. For example, ten JPEGs may be referenced by a single activity which requires a learner to sort ten cards in order of priority.

- Activities may be combined, using an appropriate sequencing or orchestration specification, to create a single composite activity.

- Tasks (each task wrapping a single activity) may be combined to create a single course, a composite set of learning experiences which are linked to explicit learning objectives.

- Resources, activities, tasks and courses may all be contained within one or more catalogues. These represent simple lists which are predominantly used for administrative and commercial processes. Catalogues may be nested within each other and may be associated with product information (i.e. access and rights).

- A number of physical files which require distribution can be aggregated into a single package. It is the intention of the SALTIS ICIG to create an interpretative framework which will allow resources (both atomic and composite) and activities (both atomic and composite) to be included in current forms of content package. In future, it is envisaged that tasks may also be contained within packages.

While the <course> represents a form of pedagogical aggregation, the <catalog> represents a form of commercial aggregation. When a user purchases a shrink-wrapped product or subscribes to a content service, the scope of their purchase will be described by a <catalog>. Both <catalog> and <course> objects will be able to reference more than one package, where locally distributed files are required.

Recommendations

It would be advantageous if LAD and the ICIG can develop in ways which are, as far as possible, compatible with the LRE. However, there are particular aspects of the LRE profile which make convergence impractical at the moment.

This paper includes minimal recommendations which would allow the LRE profile of ILOX to be integrated with the ICIG model. At the same time, the ICIG should take what measures it can to ensure that the ICIG model is able to converge with the LRE in future. The ICIG can then propose continuing collaboration with the LRE to achieve a deeper level of convergence between the two models.

Modifications to the LRE required for short-term accommodation

All of the recommendations in this section depend on modifications being made to the LRE profile, as specified below.

1 Locate ILOX records on the cloud

Both LOM and ILOX records should be located at distinct URLs by which they can be identified. Where it is considered practical to serialise several records in a single XML instance, the identification of the individual record should be provided by locally unique identifiers within the XML.

The following example proposes a wrapper specification for serialising ILOX records:

If this xml were located at http://www.xyz.com/records.xml, then individual records could be referenced as, for example, http://www.xyz.com/records.xml#id_1.

```
<xml ...>
  <list>
   <item identifier="id_1">
          <ilox>...</ilox>
   </item>
   ...
  </list>
</xml>
```

Figure 6. XML example to show serialisation of ILOX records

2 Separate LOM instances from ILOX

Content packages should be able to reference LOM instances independently from the ILOX wrappers that contain them. ILOX instances should therefore include references that point to an external LOM instance, instead of including the LOM in line.

Given that ILOX provides an explicit extension point, and assuming that the LRE intends to continue to support ILOX as its versioning solution, the rest of this paper assumes that the simplest solution in the short-term is to extract the LOM instance from the ILOX wrapper and insert a reference in the ILOX to the LOM.

3 Mandatory identifiers at all levels of the ILOX hierarchy

Any level of the ILOX hierarchy which references an external LOM instance needs to carry a unique identifier, allowing the LOM instance to create a reciprocal link back to the specific element with which it is associated.

4 Navigation from LOM to ILOX

Where a LOM instance is referenced from an ILOX record, a reciprocal reference could be included in the LOM instance using a value for <relation> of <kind> "isPartOfIloxWork". The reference should include the identifier of the element to which the LOM is attached.

The LOM <relation.description> element(s) can then be used to describe the different varieties of learning objects that can be accessed through the ILOX record using the following pseudo-codes for the language field. While the <relation.kind> element shows *how* to navigate, the <relation.description> element shows *why* the user might *want* to navigate to the ILOX record.

x-ilox-v	Different versions of this object are available
x-ilox-t	Different translations of this object are available
x-ilox-v	An interactive preview of this object is available
x-ilox-p	A printable version of this object is available
x-ilox-a	Access and acquisition information for this object is available

Figure 7. Vocabubulary for LOM <relation.description.language> when <kind> equals "isPartOfIloxWork"

```
<relations>
  <relation>
   <kind>isPartOfIloxWork</kind>
   <resource>
          <identifier>
              <catalog>URI</catalog>
              <entry>http://www.xyz.com/ilox.xml#id_1</entry>
          </identifier>
   </resource>
   <description>
          <language>x-ilox-t</language>
          <string>French, Spanish, English</string>
   </description>
   <description>
          <language>x-ilox-v</language>
          <string>4.0.5</string>
   </description>
  </relation>
</relations>
```

Figure 8. XML example showing use of LOM <relation> with <kind> equal to "isPartOfIloxWork"

The example of the use of "isPartOfIloxWork" given in Figure 8 suggests the following.

- The use of a pseudo-language code is not ideal, particularly as it is then not possible to provide real translations of the free text explanatory string. However, it is the best means available in LOM for indicating reasons why the user may wish to look at the ILOX record, a purpose for which the explanatory string is not otherwise necessary. If the ILOX record is used to manage alternative, language-specific LOM instances, the use of multi-language records would not be necessary in any case.

- The existence of a <relation.description> which indicates that the current version of the learning object is deprecated assumes that the publisher is able to edit the LOM instance after its initial release. This is an advantage of ensuring that the LOM has a separate and authoritative existence on the cloud. Further work may enable interested systems to subscribe to an update service, allowing such authoritative changes to the LOM to be broadcast automatically.

- This proposal should only be included within he ICIG LOM profile if and when the LRE implements the recommendations proposed under *Modifications to the LRE required for short-term accommodation* on page 16.

Proposed features to be introduced in any case to the ICIG profiles

1 Modifications to the LOM <lifecycle.status> element

As further support to the versioning of learning objects by ILOX records, the vocabulary of LOM <lifecycle.status> (currently "draft", "final", "revised" and "unavailable") should

be supplemented by the term “deprecated”. This indicates that the LOM instance does not represent the most recent version of the underlying work (which can be accessed through the associated ILOX record), even though it is still available.

2 Navigation from Learning Object LOM to Manifest LOM

Some LOM instances referenced from the content package will correspond to learning objects below the level of the package manifest. These LOM instances cannot navigate directly to an ILOX work, which the LRE profile anticipates will correspond to a whole package.

```
<relations>
  <relation>
   <kind>isPartOfPackage</Kind>
   <resource>
          <identifier>
              <catalog>URI</catalog>
              <entry>http://www.xyz.com/manifestLom.xml</entry>
          </identifier>
   </resource>
  </relation>
</relations>
```

Figure 9. XML example showing the use of LOM <relation> with <kind> equal to “isPartOfPackage”

In these cases, the LOM instance for the contained learning object should allow navigation to the LOM instance for the containing manifest by including a <relation> record of <kind> “isPartOfPackage”, with the <resource> value containing a reference to the URL of the LOM of the manifest. In this case, <relation.description> is unnecessary.

3 Navigating from Manifest LOM to Learning Object LOM

It may be advantageous to allow repositories indexing a manifest LOM instance to navigate to all learning objects LOM instances contained within that package, without loading and parsing the package manifest. This could be allowed by the inclusion within the manifest LOM of <relation> elements of <kind> “hasPart”.

4 Navigating to the LAD metadata

The LAD architecture allows for the creation of new types of metadata which define how a learning object should be handled at runtime. The LAD object will itself contain links to the LOM instance, so that it too can be treated as a “point of entry” in an environment in which learning management systems are managing entirely cloud-based content. In this environment, the LAD metadata will be essential to the proper functioning of the learning object.

ICIG LOM profiles corresponding to individual learning objects will require that the URL of the LAD object must be provided as the primary identifier of the learning object in <general.identifier>.

Lists of LAD learning object records are called <catalog> objects. LAD catalogs may either reference or wrap LAD object records. Where a LAD object is wrapped in a <catalog>, it may be externally referenced using the "#" bookmark notation, as in "http://www.xyz.com/catalog.xml#lad_object_1".

```
<general>
  <identifier>
   <catalog>URI</catalog>
   <entry>http://www.xyz.com/catalog.xml#lad_1</entry>
  </identifier>
</general>
```

Figure 10. XML example showing use of <general.identifier> with URI resolving to LAD object record

5 Encoding of Creative Commons licenses within LOM instances

The LRE profile encodes a Creative Commons license in 6.3 <rights.description> by using a pseudo language code of "x-t-cc-url".

Although this device works, the use of pseudo-language codes is not ideal as it represents misinterpretation of the langstring type. A preferable method, which is more consistent with the need for other kinds of external reference, would be to use a <relation> with <kind> equal to "hasCreativeCommonsLicense". The <relation> would then have a <resource> element pointing to the URL of an appropriate Creative Commons license.

Handling access and acquisition

There are a number of different access control models, which are accommodated by the ICIG profile in different ways.

1 Freely available learning objects on the cloud

It is a principle of LAD that the actual resource should always be approached through the LAD object, ensuring that essential rights and runtime management information is ingested. In this case, the way to run the object is to follow the links given under *Navigating to the LAD metadata* on page 19. This means that the LOM <technical.location> element is not supported for individual learning objects.

2 Learning objects which are packaged or require purchase

Where learning objects require purchase or are packaged (in the sense of including locally installed files), the learning object cannot be acquired individually. The user wishing to negotiate access should follow the "isPartOfPackage" reference to the LOM corresponding to the package manifest.

3 Freely available manifests

Where a manifest can be acquired individually and without payment, a direct link to a URL which resolves to the package corresponding to the manifest should be included in the LOM <technical.location> element.

4 Manifests which can be individually purchased

Where a manifest requires negotiated access, information may at present be embedded within an appropriate ILOX record, as specified by the LRE profile. Given integration with the LRE profile, users wishing to negotiate access could navigate to the ILOX record, following an <relation> of <kind> “isPartOfIloxWork” which has a description attached to it of pseudo language “x-ilox-a”.

In the longer term, it would be preferable for all acquisition information to be moved to the LAD catalog.

5 Manifests which can be purchased as part of a larger product

Manifests for which access is attached to a larger product should include a <relation> link with a <kind> value of “isListedInCatalog”, pointing to a LAD <catalog> which, in turn will provide access information (in its simplest form, a link to an acquisition page). Failing integration with the LRE and/or deprecation of ILOX, packages which can be individually purchased should be referenced by their own dedicated <catalog>.

6 Content which cannot be acquired

Although it is strongly recommended (and is in the interests of most suppliers) to include access information using one of the six methods outlined above, it should not be mandatory to do so. The primary use of LOM within a packaging environment is to describe content which the user already possesses, so it should be permissible to create learning object metadata which does not include access information.

The six cases above suggest the following means of handling access.

1 Where a learning object is available on the cloud, its URL will be given in its LAD object record, supported if necessary by access information in an associated <catalog> record.

2 Where a learning object requires local files, the user wishing to negotiate access should follow the link given by the <relation> of <kind> “isPartOfPackage” to the LOM corresponding to its package manifest. The manifest’s LOM should have a <technical.location> element if the package is available, either freely or under restricted access.

   If the package is only available to users who have negotiated access, the means of doing so must be provided within a top-level <catalog>. It follows that a package manifest which:
   - has <copyrightAndOtherRestrictions> equal to “yes”, and

- has no <relation> element of <kind> “hasCreativeCommonsLicense”, and
- has provided a value for <technical.location>,
- must include at least one <relation> of <kind> “isListedInCatalog”.

3 A LOM instance corresponding to a package manifest which does not include a value for <technical.location> and does not include a <relation> of <kind> “isListedInCatalog”, does not support acquisition. Such a record should not be indexed by repository systems and search services, using LAD and LOM only.

Where a package contains content, all of which is available on the cloud, the package itself is redundant. Such content aggregations should in future be provided as composite resources, composite activities or courses, and indexed directly by repositories.

Navigation between different metadata records

Figure 11 summarises the proposed ICIG navigational links between the different metadata records associated with a single learning object. Arrow heads indicate a navigational direction: the fact that A links to B does not necessarily mean that B links back to A. Solid arrow heads show that the link is mandatory, open arrow heads indicate that the link is conditional and in these cases, an “if” statement on the diagram explains the circumstances in which the link should be provided.

The two entry points for the navigation diagram are the <catalog>, which may either be local (in the case where local files are referenced) or global. The global catalog will be necessary in either of two cases:

- all referenced content is on the cloud;
- where the content supplier wishes to provide access and acquisition instructions for locally packaged content, in which case, versions of the same <catalog> will exist in local and global versions.

The following points reference Figure 11.

1 A local <catalog> must include references to locally packaged learning objects described in local package manifests. If no locally packaged learning objects are included within the <catalog>, then a global <catalog> should be used instead. A local <catalog> may have a globally accessible equivalent in order to provide an authoritative version of all metadata, including access information.

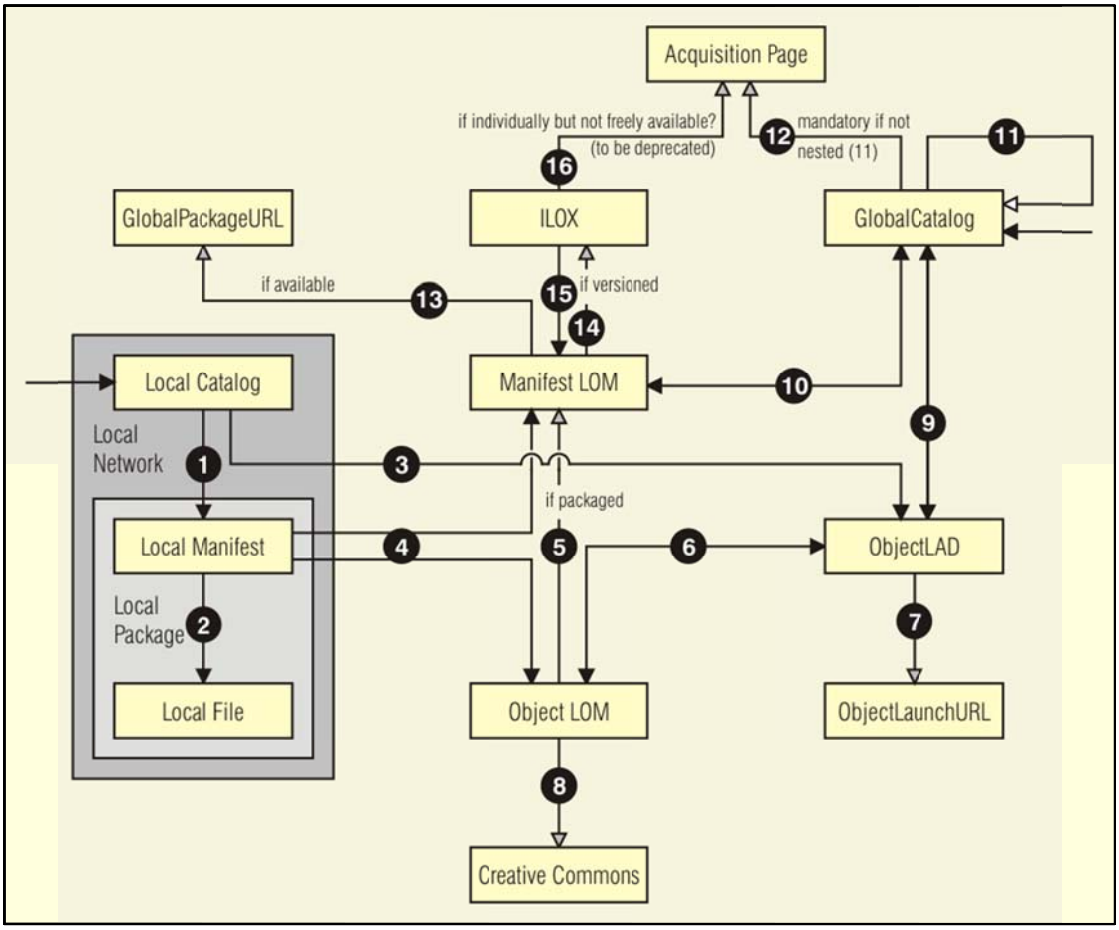
2 A local package manifest references local files.

3 A local <catalog> must include references to cloud-based LAD records for all learning objects, both locally packaged and on the cloud.

4 Package manifests must reference LOM instances, located on the cloud, at both <manifest> and learning object (either <organization> or <resource>) level.

5 All LOM instances corresponding to packaged learning objects must reference a LOM instance corresponding to the package manifest, using a <relation> of <kind> “isPartOfPackage”.

6 All LOM instances corresponding to learning objects which are referenced from a LAD <catalog> must reference appropriate LAD records using the URI which acts as the learning object identifier. All LAD records must reference a corresponding LOM instance.



7 LAD records which describe learning objects on the cloud must reference that learning object’s launch URL. Where the cloud-based learning object is not freely available, the LAD record will also indicate the need to negotiate access, normally by accessing an associated <catalog> object.

8 Where individual learning objects are freely available, they should reference an appropriate Creative Commons license using a <relation> of <kind> “hasCreativeCommonsLicense”.

9 LAD object records must reference the global <catalog> records that contain them “officially” (i.e. are produced by the publisher and are associated with access information). All <catalog> records either wrap or reference the LAD records that they contain.

10 Where it includes locally packaged content, a global <catalog> must reference the LOM instances for the manifests of the packages which contain such content; and LOM instances corresponding to package manifests must carry reciprocal links to any <catalog> that contains them, using the LOM <relation> element of <kind> “isListedInCatalog”.

11 A global <catalog> may be nested by reference inside another global <catalog>.

12 A global <catalog> which is not nested inside another global <catalog> (i.e. a top-level catalog) must reference an acquisition page, giving access information.

13 A LOM instance corresponding to a package manifest where the package is available (either freely or after negotiating access) should link to a URL resolving to a download of the entire package.

The following three links are subject to modifications being made in the LRE profile (see *Modifications to the LRE required for short-term accommodation* on page 16).

14 A LOM instance corresponding to a package manifest which has versioning information contained in an LRE ILOX record should link to that ILOX record using the LOM <relation> element of <kind> “isPartOfIloxWork”.

15 An ILOX record which provides versioning for packaged content must link to appropriate LOM instances equivalent to package manifests.

16 Where ILOX-managed content packages are individually but not freely available, the current LRE profile specifies that the ILOX record will link directly to an appropriate acquisition page, equivalent to what ILOX refers to as a “landing page”. It may be desirable to deprecate the inclusion of access information within ILOX records in future, restricting them entirely to <catalog> records.

Next steps

This model outlined in this paper will be implemented in a set of technical documents which will be produced within the SALTIS ICIG in early 2011:

- ICIG LOM profiles;
- Interpretative framework enabling the disaggregation of existing content packages, specifying the agreed representation of learning objects in VLEs;
- Learning Activity Definition, detailing <resource>, <activity> and <catalog> metadata.

Parties interested in contributing to this work should apply to join the SALTIS Interoperable Content Implementation Group.