`

Technical discussion and rationale

to support the requirements for a UK content packaging profile

Version 1.0

Crispin Weston

9 November 2009

`

Contents

`

Introduction

About this document

The *Interim Report on content publisher interviews* (hereafter, the *Interim Report*) established a range of emerging requirements for a content packaging profile in UK schools. The production of a final requirements document requires, not only the validation of the emerging requirements with other key stakeholder groups, but also the amalgamation of the various requirements into a technically coherent proposal.

It is the purpose of this technical discussion to provide the rationale for such a vision, which will form the basis of the final stage of the consultation process.

Terminology

The term "learning object" is used in this paper to refer to any reusable, atomic piece of content. It is recognised that "learning resource" is more likely to be understood by end users. This sense of "resource", used as a non-technical synonym for "learning object", should not be confused with the IMS Content Packaging node named ***`resource`***. It is for the disaggregation model produced by this project to establish the extent of the correlation between the two.

A learning object which implements the SCORM runtime is in SCORM parlance called a "Sharable Content Object" (SCO). As SCORM is currently the only protocol available which supports runtime interactions, the term "SCO" will be used when discussing the behaviour of learning objects in relation to the runtime.

A resource which does not implement the SCORM runtime is in SCORM parlance an "asset". This definition is not always helpful, bearing what may often be a misleading implication that the main purpose of SCORM assets is to provide the building bricks for SCOs. It is therefore proposed that under this profile, the definition of "asset" should be tightened. (See *Declaring* **`resource.`** on page 11).

The *Interim Report* referred to the fact that much of the terminology in this area ("learning platform", "VLE", "learning content" etc) is problematic. It may be that such any attempt to change well established usage of these terms is doomed to fail and that it may be more productive to create a new terminology, more modest in scope.

If the profile which emerged from the BECTA project were, for example, called the "UK Schools Content Packaging Profile" or "UKSCPP", then there might be merit in talking of an "UKSCPP manager" rather than a "VLE" and "UKSCPP content" instead of "learning content". Such a new terminology would have the benefit of being neutral and not conflicting with anyone's existing definition of what a VLE is.

`

As far as UKSCPP content is concerned, it is as well to restate the point that it will need to subsume both *activities* (later described as *interactive* resources) and *expositive* resources; and that these may well be delivered by remote web-services. In this case, the content package would be a medium for distributing metadata and not the resources themselves. This taxonomy of learning content will be elaborated in *Declaring* ***`resource.`*** on page 11.

Disaggregation, encapsulation, navigation and termination

A terminology for aggregation and organisation

The following terminology is proposed to help clarify the discussion of aggregation.

Term	Meaning
Learning content	Abstract term for all the digital "stuff" that can be managed by a VLE and which delivers learning. This will include static data (such as text and video) as well as applications, plug-ins, players and widgets.
Resource (technical)	An atomic piece of learning content. Although a resource may often be composed of many files, those files should be regarded (either for technical or commercial reasons) to be inextricably bound together.
Resource (non-technical)	An abstract term to describe assets, learning objects, and aggregations (see below).
Asset	A resource which is not intended to be assigned, but rather used freely in the creation of other learning objects and the completion of tasks. Clip-art is a common example.
Learning object	An individual resource which may be assigned for the purposes of learning. The term "object" blurs the distinction between a data file and the application that manipulates it[1]—"objects" may therefore include applications, services and widgets as well as static data. From the perspective of any further aggregation, it is the learning object which should be regarded as the basic atomic unit.

[1] According to Wikipedia at http://en.wikipedia.org/wiki/Object_(computer_science), software objects "bring together data components with the procedures that manipulate them". Although backed by all the technical hinterland of "object oriented programming", this blurring of data and procedure has great significance for the design of user interfaces which affect the experience of non-technical users. Many users are not even aware that they have to open a browser application to render a web page; or that in double clicking an icon representing a Word document, they are first launching the Word application and secondly loading the Word file – they just think that they are opening a single "object".

Aggregation	A collection of learning objects (and other aggregations) which can be made the subject of a single assignment.
Interstitial data	All data within an aggregation which is not defined as being part of asset or learning object. Interstitial data cannot be reused in other contexts but is bound to a particular aggregation. It can be thought of as the mortar which binds together the reusable bricks in a particular wall.
Orchestration	The process of managing the delivery of an aggregation.
Assignable unit	An atomic object or aggregation that is intended to be made the subject of a single assignment.
Assignment	The allocation of an assignable unit to a particular assignee (see page 15).
Organisation	The management of assignable units or the structure (such as a hierarchy of nested folders) within which these assignable units are organised. Note that within this proposal, the ***`organization`*** node in an IMS content package should be regarded as the root node of an aggregation and not an organisation.
Folder	A container within an organisational structure. Following the observation above, the folder will in future be represented within a content package by an IMS ***`manifest`*** node and not by an ***`item`*** node under an ***`organization`*** node.
Sequencing	A form of aggregation in which objects are launched one after another. There are two fundamental types of sequence: *flow* sequencing, in which the order of objects is determined automatically by the VLE; and *choice* sequencing, in which the basic navigational paradigm is one in which the student chooses what to do next. Note that a choice sequence may in practice appear to be very similar to accessing a folder in the VLE's organisational structure: the difference is that, where such a concept is applicable, the choice sequence is treated as a single assignment while the organisational folder is treated as a collection of different assignments.

Figure 1. Glossary of terms relating to aggregation.

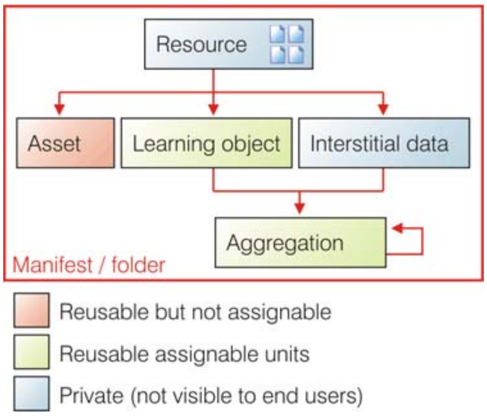

Resource
Asset
Learning object
Interstitial data
Aggregation
Manifest / folder
Reusable but not assignable
Reusable assignable units
Private (not visible to end users)

Figure 2. Diagram showing the relationship of different terms relating to aggregation.

`

Packaging is ephemeral

The *Interim Report* has established the need to clarify the disaggregation model as a key prerequisite for the later introduction of sequencing. It makes clear that disaggregation must be subject to publisher permissions and provides a preliminary description of what disaggregation will look like in the context of a VLE.

Any concept of disaggregation carries the implication that packaging is ephemeral. For any package which is capable of disaggregation, the VLE should not display links to the package but rather to the learning objects and aggregations that it contains.

The distinction between package and contents is often academic in the current environment, where most packages contain only one learning object or aggregation. This, however, will change when a robust disaggregation model is provided, as illustrated by Figure 3.

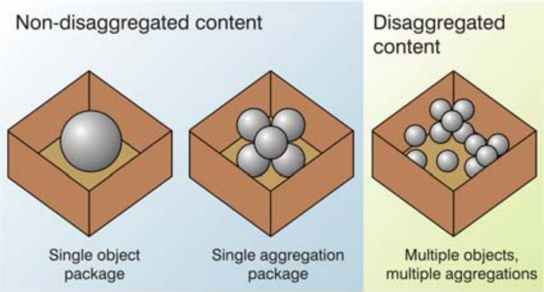

Non-disaggregated content
Disaggregated content
Single object package
Single aggregation package
Multiple objects, multiple aggregations

Figure 3. It is possible to identify content with its packaging only in models which do not allow disaggregation

The VLE will need to organise content objects in some way, perhaps within a hierarchy of folders. There is no reason why VLEs should not use the name of the package to create a default folder to contain the package contents—but teachers must be able to move disaggregated content objects into different folders as they wish, as shown by Figure 4.

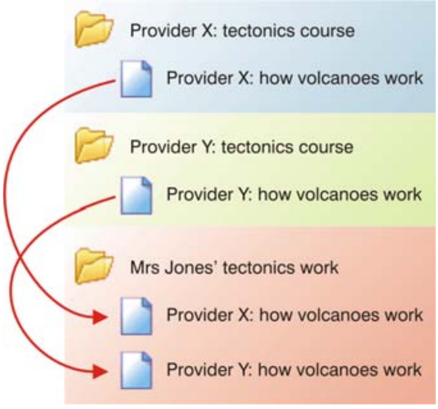

Provider X: tectonics course
Provider X: how volcanoes work
Provider Y: tectonics course
Provider Y: how volcanoes work
Mrs Jones' tectonics work
Provider X: how volcanoes work
Provider Y: how volcanoes work

Figure 4. It should be possible easily to move disaggregated learning objects within the VLE's organisational structure.

One proviso to the "packaging is ephemeral" principle is that if a future version of the profile allowed the redistribution of re-mixed content, original content would need to be redistributed in the context of its original manifest. The packaging would need to be preserved for this purpose.

Encapsulation and navigation

Given that the first version profile will not include any speci specification, the only available aggregation model will be t organization provided by IMS Content Packaging. This might be thought of as a table of contents or series of nested menus.

Even a default "table of contents" could usefully be reorganised by a teacher wanting to combine content from different sources or insert user-generated material into a commercially provided course; and even this basic level of reorganisation requires content which has been robustly disaggregated.

`

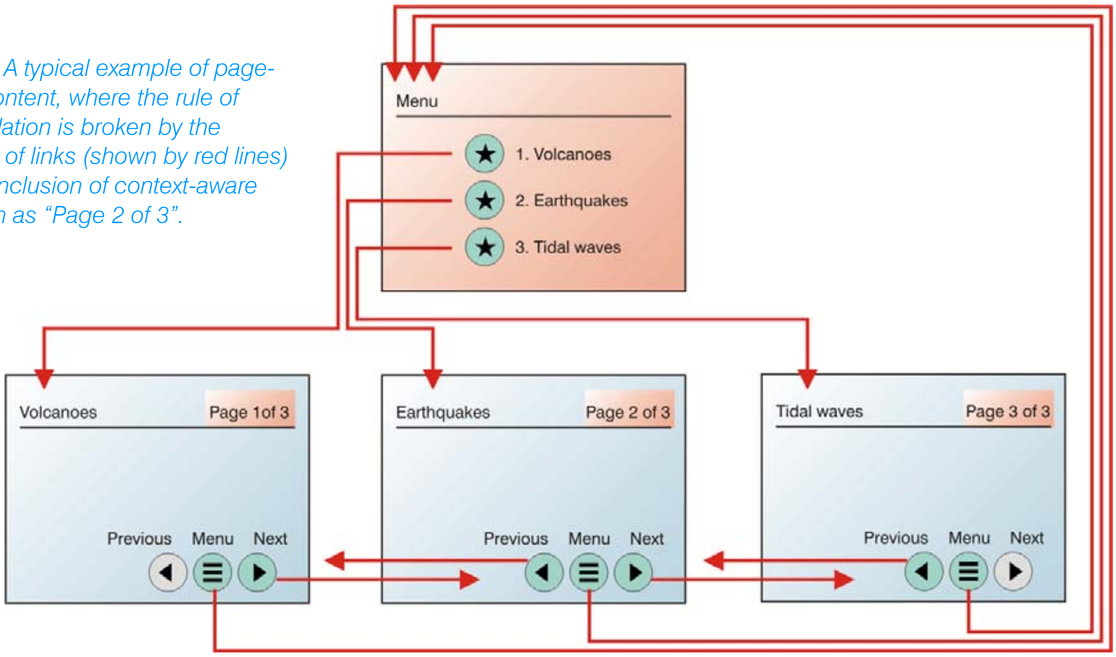


Figure 5. A typical example of page-based content, where the rule of encapsulation is broken by the inclusion of links (shown by red lines) and the inclusion of context-aware text, such as "Page 2 of 3".

Objects capable of being disaggregated must be encapsulated. That is to say that the user must not be able to navigate directly from one object to another, nor must an object make any assumptions about the context in which it is being deployed.

Figure 5 illustrates a typical example in which the rule of encapsulation is commonly broken by a straightforward organization containing multiple, expositive, page-based resources. A VLE which wished to launch only the page entitled "Earthquakes" could not prevent the student navigating away to other pages in the original series by following the *Next, Previous* and *Menu* buttons. The captions, reading for example "Page 2 or 3" would also be likely to be misleading.

Common Cartridge avoids this problem by treating the package (or cartridge) as a single entity which is not subject to disaggregation. But adoption of this approach would prevent the delivery of what has been identified by content publishers as an essential potential benefit of VLEs: the ability to re-sequence disaggregated learning objects.

SCORM offers two alternative solutions to the problem:

- allowing content to display *Next*, *Previous* and other navigation buttons conditionally, under the control of the VLE;
- replacing the navigation controls embedded within the content page with new controls (typically a hierarchical tree) displayed in a pane managed by the VLE.

The first solution requires moderately complex programming and is not being considered for the current profile.

`

The second solution is illustrated by Figure 6. The green pane is part of the VLE and contains all navigation controls. The blue pane is part of the content and, being encapsulated, displays no navigation controls. The teacher or other intermediary can now change the organisation displayed in the navigation pane, for example by inserting a user-generated quiz, without conflicting with any assumptions made by the resource about the organisation in which it is embedded.

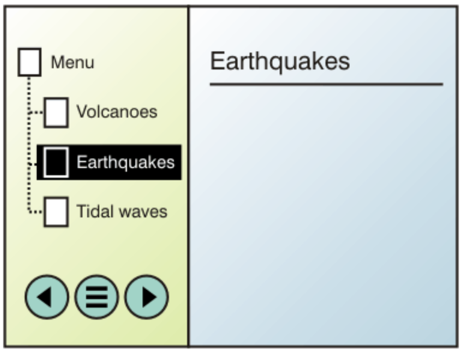


Figure 6. Running encapsulated content in a SCORM environment, with all navigation controls in the VLE-controlled navigation pane.

Although familiar to anyone who has used SCORM content, the model illustrated in Figure 6 presents a number of problems.

- The VLE's navigation pane will take up a significant part of the screen, possibly cramping the space made available to the content.
- If the content represents an activity rather than a page of information, the user may finish the activity by selecting another page on the navigation pane. This may have undesirable consequences, such as the loss of work or the premature submission of work for marking.
- The appearance of re-sequenced third-party content within a VLE frame may have implications for branding. The user may assume that poor quality content has been endorsed or is in some way associated with the VLE or with other juxtaposed content.
- The requirement to encapsulate individual learning objects (by the removal of hyperlinks) may degrade the content or even render it unusable outside the VLE.

These issues are discussed in the following sections.

The significance of calling *Terminate*

The distinction between activities and expositive resources mirrors the distinction made in SCORM between a "Shareable Content Object" (SCO) and an "asset". The technical difference between a SCO and an asset is that a SCO implements the SCORM runtime, while an asset does not.

The runtime allows a set of interactions between a SCO and a VLE-implemented API which can be used to pass data between VLE and content while the content is running. This data exchange is managed by the SCO, which makes a series of calls to the `GetValue` and `SetValue` functions, implemented by the VLE-provided API. The conversation is started by the SCO calling the `Initialize` function and is finished when the SCO calls `Terminate`.

The only mandatory calls in the SCORM runtime are those to `Initialize` and `Terminate`. Implementing the runtime at such a rudimentary level appears to do nothing but open a data pipe down which no data is passed; but this apparently empty shell does achieve at least two things:

```
API.Initialize; //start the session
…
score=API.GetValue(“cmi.score.raw”); //read previous score from VLE
…
/*modify the “score” value as a result of student interactions*/
…
API.SetValue(“cmi.score.raw”, score); //write score to VLE
…
API.Commit;  //all data changes are saved
API.Terminate; //end the session
```

Figure 7.Truncated example of a set of calls to the VLE-provided API, which might be embedded in a content page.

- if the content successfully discovers the API and calls `Initialize`, it can assume that it is running in a managed environment under the control of a VLE and adapt its behaviour accordingly;
- calling `Terminate` tells the VLE when the content has finished running, allowing the VLE to react appropriately.

This second point is a key prerequisite for automated “flow” sequencing, in which the student is led through a learning process without having constantly to click a “Next” button.

The second point above also makes clear the correspondence of a SCORM SCO to an activity. It makes sense to talk of an activity “finishing” because the activity has been completed, because the actor has chosen to stop the activity, or because of some sort of time limit; even though it makes sense to talk of “finishing” a page of text, there is a problem in that the text does not know when it has been finished; and it does not normally make sense at all to talk of “finishing” a resource such as an image.

The call to `Terminate` a runtime session (which is likely to occur in any runtime environment and not just in SCORM) is therefore significant as a prerequisite both for activity-based learning and for flow sequencing.

Automated encapsulation

This term is suggested to describe a learning object which can automatically encapsulate itself, by hiding links and context-dependent information when it detects that it is running in a managed environment.

All that is required is for the content to determine whether it is running in a managed environment. This can be achieved under the SCORM runtime by searching for a VLE-provided API and calling ***`Initialize`***. Inappropriate buttons and links can then be hidden by using code along the lines of that shown in Figure 9.

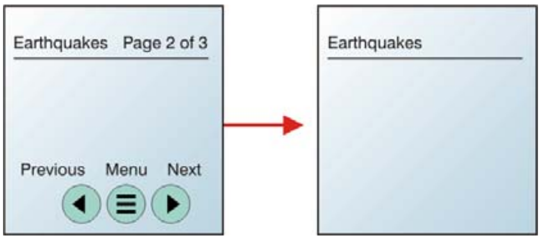


Figure 8. Encapsulation requires the removal of links and context-dependent information which is not managed by the VLE.

```
if (API exists) and (API.Initialize() returns “true”) then
     set api_started to “true”
else
     set api_started to “false”;
if (api_started equals “true”) then
begin
     hide previousButton;
     hide nextButton;
     hide menuButton;
     hide pageReference;
end;

/*run activity*/

if (api_started is true) then
     call API.Terminate();
```

Figure 9. Pseudo-code (which should be comprehensible to non-technical readers) which would allow a piece of content to encapsulate itself if it detected that it was running in a managed environment.

`

As well as being run outside a VLE, the *Earthquakes* page can also be embedded within a SCORM asset called *Tectonics*, the entry point of which is provided by the top level pink *Menu* page shown in Figure 5. If the Menu page is declared as a SCORM asset, the VLE will not provide it with any runtime API and all the links shown in Figure 5 will operate normally.

```
<manifest …>
    <organizations …>
        <organization identifier=”PUBLISHER_INTERFACE”> …
            <item identifierref=”TECTONICS”…/>
        </organization>
        <organization identifier=”VLE_INTERFACE”>…
            <item identifierref=”VOLCANOES”…/>
            <item identifierref=”EARTHQUAKES”…/>
            <item identifierref=”TIDAL_WAVES”…/>
        </organization>
    </organizations>
    <resources>
        <resource identifier=”VOLCANOES” scormType=“sco”>…</resource>
        <resource identifier=”EARTHQUAKES” scormType=“sco”>… </resource>
        <resource identifier=”TIDAL_WAVES” scormType=“sco”>… </resource>
        <resource identifier=”TECTONICS” scormType=“asset”>…
            <dependency identifierref=”VOLCANOES”/>
            <dependency identifierref=”EARTHQUAKES”/>
            <dependency identifierref=”TECTONICS”/>
        </resource>
    </resources>
</manifest>
```

Figure 10. Provision of expositive content in a polymorphic form, behaving either embedded assets or encapsulated SCOs.

The outline XML shown in Figure 10 shows the declaration of the package in which the dynamic resources *Volcanoes*, *Earthquakes* and *Tidal waves* are all declared as SCOs (purple) and are accessed flexibly from a VLE managed organization (blue); but they are also accessible from the SCORM asset Tectonics (referenced from the red organization), when they will behave as interlinked, embedded assets, as shown in Figure 5. Note the important declaration of the dependency of the Tectonics resource on its consituent pages (green).

This scenario represents important functionality for publishers. The presentation of content under a SCORM navigation pane often represents, from the publisher’s point of view, a degradation of the user interface. It is therefore desirable for publishers to be able to present their content, by default, embedded within their own navigational interface; yet allow constituent parts to be remixed using the VLE’s generic navigational interface.

`

In order for this method to function, it is important to stress that the VLE must not make available a runtime API to a resource which has declared itself to be of *`scormType="asset"`*.

The initiation of the *`Terminate`* function

The page which manages its own automatic encapsulation, as illustrated in Figure 9, implements the runtime for the purpose of discovering whether it is running in a managed environment. It might also implement the runtime in order to download information about the current student which would enable it modify its presentation.

In either of these scenarios, the page is required to call the *`Terminate`* function even though, being an expositive resource and not an activity, it has no way of telling when the student has finished reading the page.

The page could allow self-termination by providing the student with a button which might be labelled "continue" or "done". Note that the fact that the button calls *`Terminate`* does not mean that it will necessarily produce an effect that would justify a label of "exit".

Where the page was embedded within a flow sequence, such as shown in Figure 11, the student could progress from one object to the next by pressing the *Done* button (or by finishing the activity, if there is one). Each time the *`Terminate`* function is called by the content object, the VLE will progress automatically to the next object. One incidental advantage of this scenario is that the content can occupy the whole screen, there being no requirement for a VLE-provided navigation panel.

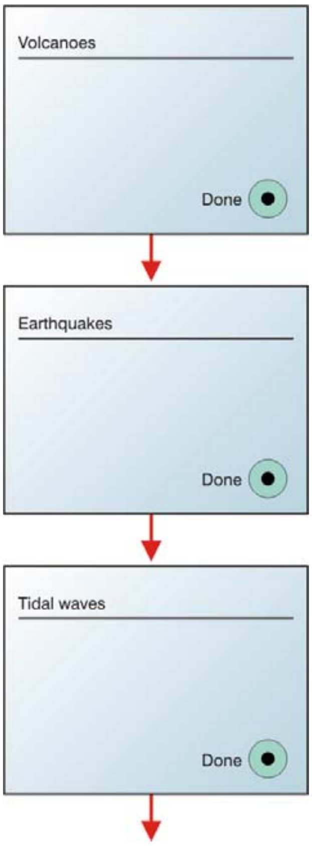


Figure 11. Provision of a "done" button might make sense for activities or for objects embedded in a flow sequence.

It should also be noted that, for sake of a smooth transition, the VLE is likely to want to load the new content into the same window as used by the last content object. A content object operating under the runtime should never attempt to close the window in which it is displayed. The call to *`Terminate`* is all that is required to finish the session.

Even when flow sequencing becomes available in a future version of the profile, it will not always be appropriate to display a *Done* button. In the *choice* organisation illustrated in Figure 6, the *Done* button would be redundant. Clicking the button would not change the status of the page or provide sufficient information to the VLE to enable it to navigate away from the page.

In circumstances where a content object cannot initiate a call to the *`Terminate`* function itself, this call is made from within the page's *`onUnload`* event. This is called automatically when the page is unloaded, perhaps as a result of the student navigating away from the page by interacting with the VLE's navigation panel.

```
<html>
     <head>
          <script language="JavaScript">
               function unloadPage() {API.Terminate();}
          </script>
     <head>
     <body onUnload="unloadPage();">
     //…
     </body>
</html>
```

Figure 12. Simplified code to call Terminate() automatically when the page is unloaded.

Declaring *`resource.technicalResourceType`*

The discussion above makes an important distinction between those interactive SCOs which can initiate their own termination and those which do not; and has explained why the VLE will need to treat these two types of objects differently in the context of different types of sequencing. It follows that it would be more useful to think of there being three levels of runtime implementation and not just the two (SCO and asset) defined by SCORM.

Figure 14 shows these three different types of learning object cross referenced against two types of sequencing: *choice*, illustrated in Figure 6, which expects termination to be handled by the VLE; and *flow*, illustrated in Figure 11, which expects termination to be handled by the learning object.

Figure 14 identifies three issues, highlighted in pink. Issue 3, which will be faced by both dynamic and static resources when used with flow-sequencing or single-object assignment, can be easily resolved by the provision by the VLE of a navigation pane showing a single "done" button (see Figure 13).

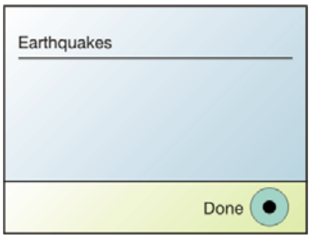


Figure 13. Provision of a "done" button by the VLE when deploying resources with flow sequencing or single-object assignment.

Issues 1 and 2 will be faced by interactive resources which are used with choice sequencing in the first version of the profile. The following recommendations would address the problems.

`

	`resource.technicalResourceType`			
	Interactive	Dynamic	Static	Asset
Explanation	Implements the runtime and initiates its own termination	Implements the runtime but does not initiate its own termination	An assignable resource that does not implement the runtime	A non-assignable object which is intended for use in building other resources or completing tasks (e.g. a JPEG).
SCORM term	SCO		Asset	
Choice sequencing	1. The requirement for a VLE navigation pane may cramp the display of immersive environments; 2. The user may navigate away from a resource when it has not finished running.	No special issues.	No special issues.	Not applicable. Being non-assignable, assets may not be included in either choice or flow sequencing. They will be subject to other forms of orchestration in the future.
Flow sequencing	No special issues.	3. May not have the means to initiate its own termination, which is required for forward progress.		

Figure 14. Issues faced by different types of learning object with different kinds of sequencing. A fourth term, "asset" distinguishes non-assignable objects which are intended for use in the creation of other content objects or in the completion of student tasks. "Dynamic" and "Static" resources may be described collectively as "Expositive" resources.

- It is proposed that a new element or attribute called `technicalResourceType` should be added to the `resource` element making the distinction between interactive, dynamic and static resources. A fourth term "asset", is proposed to distinguish non-assignable objects which are intended for use in the creation of other content objects or in the completion of student tasks.
- When running an `interactive` resource with *choice* sequencing, any navigation pane should be disabled or preferably removed until the activity has finished running.
- The content author should bear in mind that in *choice* mode, the `interactive` resource may continue to be displayed after it has finished running, and should therefore display a "finished" interface with all controls disabled. It is not recommended that the activity in this state should show a "replay" button as this is a function that will need to be controlled by VLEs in future versions of the profile.
- While the VLE should not automatically navigate away from a finished `interactive` resource in *choice* sequencing mode, the VLE may wish to indicate that the activity has finished by greying out or disabling the corresponding choice in the navigation pane.

`

Handling unexpected termination

By declaring itself to be an `interactive` resource according to the table in Figure 14, an object declares not only that it can, but also that it prefers to initiate its own termination. However, it cannot be guaranteed that this preference will be honoured and that the resource will not be closed unexpectedly. There may be cases where the VLE needs to close an interactive resource as a result of a time-out or a perception that the resource has crashed. The user may also close a content window by clicking on the *Close window* button on the browser title bar or by pressing Alt-F4 in Windows.

It is therefore desirable that all SCOs, even when they anticipate initiating their own termination, should also call the `Terminate` function from within an `onUnload` event, as illustrated in Figure 12.

The multiplicity of ways in which a resource may be closed, and the *Terminate* function called, leads to a number of technical complications in the VLE implementation. The VLE cannot unload the content as part of the execution of the `Terminate` function, as the `Terminate` function may itself have been called by the content targeted for closure. The `Terminate` function may be called twice: once as a way of initiating unloading and a second time in the course of unloading. The VLE should probably ignore the second call to `Terminate` without throwing any error condition which might disrupt the unloading of the resource.

Guidance should be provided to VLE providers in the implementation of the `Terminate` function and to help test their implementations and ensure that they handle all anticipated ways in which a SCO might be terminated.

Branding and “reputational distance”

It was proposed by the Interim Report that content packages should include thumbnails and corporate logos.

In this discussion, it is proposed that the term “thumbnail” should subsume two different kinds of image:

- a multi-resolution icon, targeted at students;
- a re-sizable “preview”, with an original size of 300 x 200 pixels.

In student views, the VLE would always interpret “thumbnail” to mean “icon” but in teacher views, the VLE would be free to use either an appropriate icon or the preview screenshot, appropriately sized.

The consensus amongst content publishers is that the appropriate display of corporate logos with the VLE is useful but should not be too prominent. Consistent with this view, it is proposed that VLEs should display corporate logos:

`

- alongside results returned from a search interface;
- in an *About…* dialog or screen which the user has chosen to open, alongside full copyright information.

Content publishers are of course free to include whatever branding they like within their own content and rules preventing the disaggregation of content beyond the publisher-declared learning object should help prevent any unauthorised removal of such branding.

The question of branding also needs to be considered in the light of the sequencing scenarios described above.

- Where operating in *choice* mode (see Figure 6), the branding of a piece of content may affect the student's choices. It may therefore be useful to display at least thumbnails in the navigation pane.
- Where content from different sources may be juxtaposed, there may be need to create "reputational distance" between VLE and publisher, and between different publishers.

The question of "reputational distance" is a related to the concern of some VLE suppliers about displaying content which might be perceived to be of poor quality and which might reflect badly on the platform. This concern is likely to become more acute in a plug-and-play environment where VLEs cannot exert quality control (incompatible with open interoperability) on the content being run on their platform.

The same concern that has been expressed by VLE suppliers might affect publishers in the kind of flow sequence shown in Figure 11, where branded content might be seamlessly followed by unbranded, poor quality content, which the user might naturally assume was produced by the same company as had produced the previous object to be displayed. The inclusion of branding embedded within the content page might not protect the publisher from juxtaposition with unbranded content.

The issue is further complicated by:

- aggregated content, where contributions may have been made by several companies or independent authors;
- the potential for the juxtaposition of VLE and content branding on the same screen, creating confusing messages and possibly being taken to imply that one endorses the other.

The following proposals are made to address these issues.

- Where available, thumbnails should always be displayed next to the title of a learning object.
- As well as thumbnails and where available, corporate logos should be displayed alongside full copyright information on an *About...* display.
- As well as thumbnails and where available, corporate brands should be displayed next to all results returned from product searches.

`

- No VLE branding should be displayed in any VLE interface which is displayed concurrently with running content. This avoids implications of endorsement, the clash of branding images, and has the subsidiary advantage of maximising screen space for running content.
- Where a VLE navigation pane is visible, it should include a learning object “badge” which should include:
  - the learning object thumbnail;
  - the short title;
  - the short copyright notice (see
  - *LOM Rights* section on page 35);
  - all of the above acting as one-click links to an *About…* display.
- This badge should also be displayed whenever the learning object is selected within a teacher-facing VLE interface, with the difference that in a teacher-facing interface, the badge should also include the short description (see page 29).
- Where no VLE navigation pane is visible and a sequence of content changes from one publisher to another, authoritative title and branding, the VLE should display the learning object badge as a splash screen for three seconds. Note that this requirement will not appear in the first iteration profile, which will not include flow sequencing.
- Where content is launched directly from user-created links, the original student badge should be displayed automatically when the mouse hovers over the user-created link.

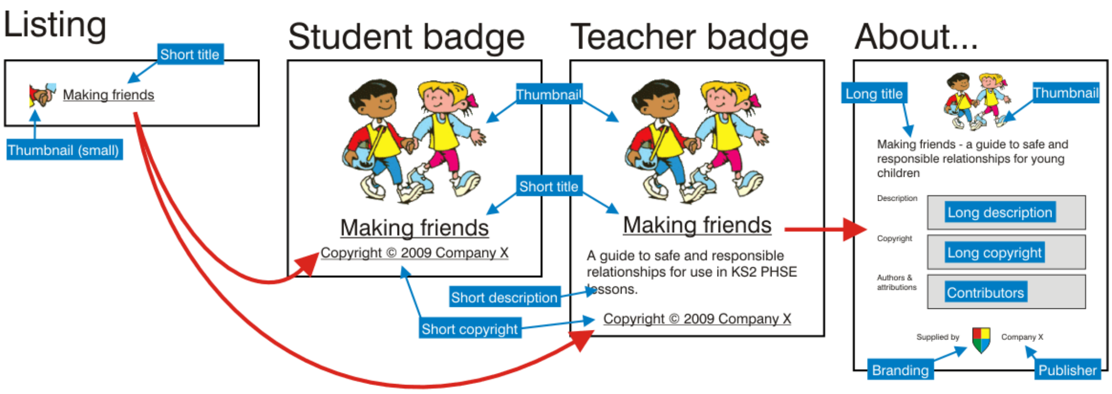


Figure 15.Use of descriptive metadata and banding in learning object listing, badges and About… display. Appropriately scaled previews can be used in place of icons in any teacher interface.

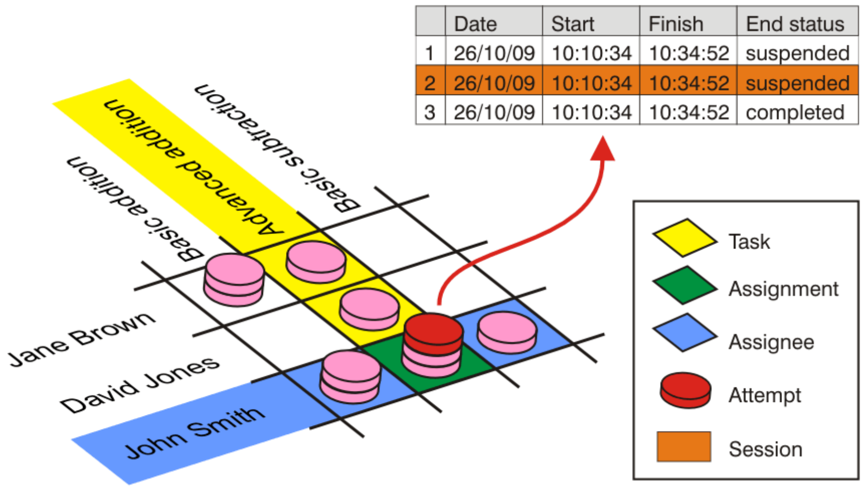
The SCORM runtime

Terminology of assignments

Before discussing the SCORM runtime, the following definitions are offered to help avoid misunderstandings in the terminology.

`

Term	Meaning
Assignments	
Task	A unit of work. Synonymous with the AICC term, “assignable unit”; and very closely related to “learning object” (though there may be some types of learning object, like assets, which are not assignable and it is conceivable that one learning object might spawn a number of tasks). In a conventional grade book, each column corresponds to a task.
Assignment	The act of making a task available to an assignee with some level of expectation that the task will be performed. A single task may be assigned to the same assignee multiple times in different contexts (e.g. formative and summative). In a conventional grade book, each cell corresponds to an assignment.
Assignee	The person or group of people to which an assignment is made. Where a task is assigned to a teaching set, each *member* of the set is, individually, an assignee. Where a task is assigned to a group to execute collectively, then the *group* is the assignee. In a conventional grade book, each row corresponds to an assignee.
Attempt	A single execution of a task by a particular assignee. The assignee may make multiple attempts at a single assignment.
Session	A continuous period during which a task is executed. An attempt may comprise multiple sessions at the end of which state can be saved to allow the assignee to resume the suspended task in a subsequent session.



	Date	Start	Finish	End status
1	26/10/09	10:10:34	10:34:52	suspended
2	26/10/09	10:10:34	10:34:52	suspended
3	26/10/09	10:10:34	10:34:52	completed

Figure 16. Representation of task, assignment, assignee, attempt and session

`

Session outcomes	
Abandon	Any data from the session is deleted. In the case of the first session of an attempt, the attempt is also abandoned and no data is preserved.
Suspend	The attempt may be resumed at a later time, any state data required having been saved on the VLE.
Finish	The attempt is closed so that it cannot be resumed in a later session and final version of any data is recorded.
Complete	The attempt has finished because it is perceived by the assignee or the learning object that the execution of the task requires no further activity. Note that a finished attempt may not be completed: a student who runs out of time on an exam has finished the exam even though he has not completed it.
Progress	A measure of the amount of activity performed in a task by the assignee as a percentage of the amount of activity that is likely to be required in order to achieve completion of the task.

Figure 17. Glossary of terminology for assignments and session outcomes.

Runtime profiles

The *Interim Report* has proposed the inclusion of the following runtime fields:

- headline scores,
- state data,
- initialization data.

The *Interim Report* also covered

- the need to record data against multiple attempts,
- the need to provide backwards compatibility with SCORM 2004 .

The following table lists all the fields in the CMI data model and suggests the grouping of fields into the following profiles.

	For publishers	For VLE suppliers
▩	Not supported	
🟦	Required for SCOs which report scores	Mandatory
🟨	Required for SCOs which save state	
⬜	Optional for SCOs	

Supported elements and vocabulary terms are shown in bold. Vocabulary terms not shown in bold are not supported.

`

Field	Vocabulary	Comment
comments_from_learner		Student-teacher communications were not rated highly in the publisher interviews
comments_from_lms		
completion_status	***"not attempted"*** ***"incomplete"*** ***"completed"*** *""*	Required to support the interpretation of results data. See *Completion Status* on page 18.
completion_threshold		See *Completion Status* below.
credit		Redundant. The VLE can discard or summarise performance data when an attempt is closed without informing the learning object.
entry	***"ab-initio"*** ***"resume"*** *""*	Supports the saving of state data by informing the learning object whether, on launch, it needs to reload state data.
exit	*"time-out"* ***"suspend"*** *"logout"* ***"normal"*** *""*	*"suspend"* and *"normal"* are required to support the saving of state data, informing the VLE, on termination of a learning object, whether the VLE should save state data. Re. *"time-out"*, it is not proposed to support time limits in the first version of the profile. Re. *"logout"*, see *Use of cmi.exit="logout"* below.
interactions		There may be significant interest in recording a log of student interactions in the future but this feature is not proposed for the first version profile.
launch_data		Proposed for inclusion in the first version of the profile—see the *Interim Report* (Teacher adaptation of content on page 32) for a full explanation of this rationale.
learner_id		See *Identifying the student* on page 26.
learner_name		See *Identifying the student* on page 26.
learner_preference		Can be implemented more flexibly by *launch_data* or in future by shared data fields.
location		Duplicates *suspend_data* or *launch_data*.
maximum_time_allowed		Time limits are not included in the first version.
mode		Can be implemented more flexibly by *launch_data*.

`

objectives		Not implemented in the first version.
progress_measure	0..1	Useful if the teacher is to make sense of scoring data, where a low score may be attributable to a lack of progress rather than poor performance. See *Completion Status* below.
scaled_passing_score		No pass marks are implemented in the current profile.
score	***scaled*** (-1..1) ***raw*** (0+) ***max*** (0+) *min* (0+)	The ability to pass score data to the VLE is rated as the publishers' top priority. The identification of the maximum number of marks which the student has been able to score in the current attempt (max) is important to any interpretation of the raw marks (raw). There seems to be little requirement to record the minimum possible score, which will in nearly all cases be 0. Values returned must be consistent with the value of *scoreReturned*. See *Use of* ***score.max*** on page 21.
session_time		This is straightforward to implement and supports proposals for the accurate profiling of *typicalLearningTime* data (see page 35).
success_status		No pass marks are implemented in the current profile
suspend_data		Required to implement the saving of state.
time_limit_action		No time limits are implemented in the current profile.
total_time		This is straightforward to implement and supports proposals for the accurate profiling of *typicalLearningTime* data (see page 35).

Figure 18. Elements of the CMI data model (used in the SCORM runtime).

Empty string return values

It is proposed that the empty string return values allowed by SCORM 2004 for the *entry, exit* and *completion_status* fields should not be supported in an attempt to reduce the potential for undefined behaviours.

Completion status

SCORM provides a runtime field, *completion_status*, with a vocabulary of *"not_attempted", "incomplete", "completed"* and *""* (undefined).

`

There is an obvious (see Figure 19) and less obvious (see SCORM documentation) mapping between *completion_status* and *progress_measure*.

progress_measure	***completion_status***
0	*not_attempted*
0 < x < 1	*incomplete*
1	*completed*

Figure 19. Intuitive mapping between completion_status and progress_measure.

The complication added by SCORM lies in the addition of the *completion_threshold* field. The idea behind this element is that if *completion_threshold="0.8",* then when *progress_measure="0.8"* an attempt would be treated as having been completed, even though not all possible activity had been performed.

It is proposed that *completion_threshold* should not be supported and a value of *progress_measure="1.0"* should be interpreted as a student having completed not all of the activity that was possible but of all the activity that was required. In future versions of the profile, it will be possible for the teacher to set this implicit "completion threshold" by editing initialisation parameters (which might allow, for example, the selection of 8 out of 10 questions). This would be more flexible and intuitive than any process which used *completion_status*, the interpretation of which would always be somewhat obscure.

How a SCO calculates a *progress_measure* which is more than 0 and less than 1 will depend on the internal implementation of the SCO. Some SCOs, due to the nature of the activity that they represent, may not be able to make this calculation.

For this reason, it is proposed that SCOs which return scores should be required to return a value for *completion_status* but only recommended to return a value for *progress_measure,* and then only when *completion_status="incomplete"*. It would be superfluous to return any value for *progress_measure* when *completion_status="not attempted"* or *completion_status="completed"* as a value can in these cases be calculated according to the mapping shown in Figure 19.

The possibility that a task might be marked as completed before all available activity has been undertaken might have implications for the interpretation of marks data. See *Use of* score.max on page 21.

Support for multiple attempts

The ability to record scores for multiple attempts requires:

`

- a clear indication of when an attempt should be finished or closed—this occurs when a SCO calls `Terminate` having set *`exit="normal"`* and *`completion_status="completed".`*;
- an ability to avoid the accumulation of worthless data by abandoning attempts—this happens when a SCO calls Terminate having set *`exit="normal"`* and *`completion_status= "not_attempted".`*

Use of *`cmi.exit="logout"`*

SCORM 1.2 allowed a value of *`"logout"`* for the `cmi.exit` field, the meaning of this value being that the student wished to exit the whole learning activity and not just to signal the termination of the current learning object. The significance of this distinction is made in *The Initiation of the* `Terminate` *function* on page 6. Without the ability for the learning objects illustrated in Figure 11 to signal the student's desire to "exit all", it would be impossible for the student to halt the flow sequence.

There are several relevant considerations:

- This is not a requirement for the first version profile, which will not include flow sequencing.
- For an unsequenced, single-object assignment, *continue* and *exit* will have the same effect. To prevent unnecessary duplication of controls, it will in future versions of the profile be desirable for the SCO to be able to discover from the VLE whether the second button was required.
- In the context of choice sequencing, the ability to *exit all* from within the SCO would be desirable but not essential. The alternative, discussed on page 6, is that the interactive resource should terminate, after which it will display a "finished", disabled interface, when the VLE navigation interface will reappear, allowing the student to exit the sequence from a button on this pane.
- The value of *`"logout"`* has been deprecated in SCORM 2004 in favour of a new navigation specification; but it is arguable that the ADL navigation specification may not be appropriate to use in UK schools. It would therefore be useful to try and co-ordinate a simplified "exit all" strategy with the ADL for future versions of the profile.

The conclusion of this discussion is that *`cmi.exit="logout"`* should not be supported in the first version profile—though it is as well that implementers are aware of the issues in this area, and of the likely need to introduce *exit all* functionality in a future version of the profile.

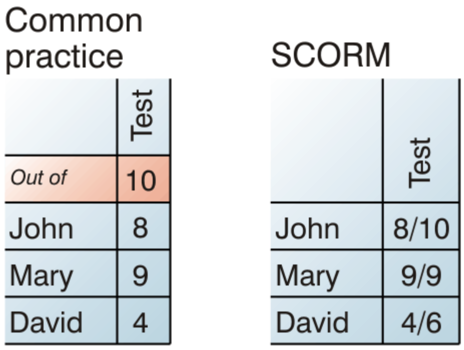
Use of *`score.max`*

Most SCOs which wish to return scores will choose either to return a scaled score (percentage), or raw marks, providing a maximum marks or "out of" suffix (as in "6 out of 10").

SCORM does not currently allow a SCO to declare the total number of marks that it is in theory capable of awarding (what the test is out of). SCORM supports the `cmi.score.max` field, which gives the maximum number of marks that a student would have been able to score on the current attempt. If a test is out of a total of 20 marks, a student who is half-way through that test would record `cmi.score.max="10"`.

This raises two issues:

- If a student has scored 5 out of the 10 available on a suspended attempt at a SCO with a maximum of 20 marks, it is unclear whether this interim score be reported as 5 / 10 (50%), or 5 / 20 (25%).
- In the digital mark-book shown in Figure 20, the VLE has no way of knowing the total marks value which would normally be displayed at the head of an assignment column (highlighted in red). Each student might instead have a different reported value for `cmi.score.max`.



Common practice

	Test
Out of	10
John	8
Mary	9
David	4

SCORM

	Test
John	8/10
Mary	9/9
David	4/6

Figure 20. A comparison of the common way in which most teachers would expect to represent a mark book, and the way in which SCORM drives mark book layout.

It is proposed that the first version profile should provide an extension to the metadata within the content package to allow SCOs to declare the maximum number of marks which may commonly be awarded by any SCO which awards marks at all.

There are two complications associated with this proposal:

- In some cases, the maximum marks available on a completed attempt might genuinely be variable or the very concept of maximum marks may be inappropriate.
- In future versions of the profile, teachers will be able to edit the initialization data of a SCO in order to allow teacher adaptation of content (see Figure 38 on page 34 of the *Interim Report*). In this case, any adaptation of the initialization data may well alter the total number of marks available because, for example, the teacher had chosen to use only 6 out of 10 questions. Any process to allow the editing of initialization parameters will therefore also have to allow for the modification of the absolute maximum marks value.

Given the declaration of absolute maximum marks in the metadata, the `cmi.score.max` is only strictly required:

- when the concept of maximum marks is appropriate and

`

- where the SCO is incomplete or where the absolute maximum mark has been declared as variable.

In practice, however, it is recommended that SCOs should return a value for *score.max* on completed as well as on incomplete SCOs. Where a value is given to the absolute maximum mark, *score.max* should never be more that the absolute maximum. The use of *score.max* when the SCO is completed allows the VLE to check for errors.

It is therefore proposed that a new metadata item, *scoreReturned*, should be created under the *resource* node, with an attribute of *scoreType*, this attribute having a vocabulary as shown in Figure 21. Where the *scoreReturned* are given as *scoreType=“raw”*, then the element should provide for an integer to specify the maximum number of raw marks which will be returned. In the first version profile, the *scoreReturned* element should have a multiplicity of 1 but in future versions of the profile, the multiplicity of *scoreReturned* could be extended to allow the reporting of multiple marks.

Proposed vocabulary for ***scoreReturned.scoreType***		
Term	Explanation	Runtime fields to be supported
raw	SCO returns score as raw marks out of a absolute maximum, which must then be declared as an integer.	*score.raw* *score.max*
variable	SCO returns score as raw marks out of a variable total marks value.	*score.raw* *score.max*
unlimited	SCO returns score as raw marks with no absolute maximum (as in “how many mushrooms can you find in 5 minutes”)	*score.raw*
scaled	SCO returns score as a floating point percentage. SCO may also return *score.raw* and *score.max* data when suspended. So long as the SCO is suspended, these values should be stored for later retrieval but they should not be displayed and can be discarded when the SCO finishes.	*score.scaled*

*Figure 21. Proposed vocabulary for **scoreReturned.scoreType attribute**, cross-referenced against the runtime elements which should be supported, depending on the values given.*

Use of *score.scaled*

Where a SCO wishes to return a scaled (percentage) score, it is recommended that for suspended SCOs, this scaled score should represent a predicted final scaled score. In other words, an interim scaled score, returned on a suspended SCO, should be calculated as a percentage of the maximum marks available to the student; while in a finished SCO, it should be calculated as a percentage of the absolute maximum marks, even if, as a consequence of not completing the SCO, fewer marks than the absolute maximum were available to the student. For example:

	Field name	Case 1	Case 2	Case 3
Given	*`scoreReturned`*	20	20	20
	`progress_measure`	0.5	0.5	-
	`exit`	*`“suspend”`*	*`“normal”`*	-
	`score.max`	10	10	25
	`score.raw`	5	5	-
Then	*`score.scaled`*	0.5	0.25	ERROR!

Figure 22. Three use-cases showing the calculation of `score.scaled` from raw marks, depending on whether an incomplete SCO is suspended (case 1) or finished (case 2); and showing the potential for an error where `score.max` is more than the value for `scoreReturned` given in the resource metadata.

Support for fractional marks

SCORM 2004 allows for floating-point numbers to be used for raw marks. This dates back to earlier versions of SCORM which did not include the floating point element, *`score.scaled`*. While floating-point numbers are clearly necessary for scaled (percentage) scores, there may be merit in restricting the values for *`score.raw`* and *`score.max`* to whole integers.

Deprecation of *`score.min`*

SCORM allow SCOs to return data not only for score.max but also for *`score.min`*. It is proposed that this element should not be supported on the grounds that in virtually all cases, *`score.min`* will be *`“0”`* and performance data in which students score a minimum mark by default is a misleading representation of student performance and is therefore of questionable pedagogical value.

Handling errors in descriptions of runtime behaviour

It may be objected that the potential for errors (as shown in the final column of Figure 22 above) constitutes a flaw in the proposal.

While most object metadata currently provides human-readable descriptions, rights data, and terms to support search and discovery, it is arguable that a lacuna in the current SCORM specifications’ ability to deliver reusable, interoperable content lies in the lack of metadata which describes a SCO’s runtime behaviour. Any VLE needs this kind of information if it is to manage a SCO intelligently. It is anticipated that future versions of the profile will include substantial extensions to object metadata which describe runtime behaviour.

Until now, the only metadata which describes runtime behaviour is *`scormType`*, which may be declared as being either of type *`“sco”`* or of type *`“asset”`*. This

`

attribute is subject to exactly the same kind of error as the ***scoreReturned*** element proposed above: a publisher may mark a resource as being a SCO when it does not in fact implement the runtime.

Any metadata the intention of which is to declare an object's runtime behaviour is subject to objective errors. These are not errors that can be avoided: they are errors that need to be managed.

There are a number of ways in which that management may be handled:

- While content may in the future be adapted, re-aggregated, and redistributed, it is important that metadata that describes objectively the object's runtime behaviour should never be changed by an unauthorised person. The first version profile should therefore establish clear guidelines on which parts of a content package and its associated metadata are public and editable and which parts are private and read-only.
- In some but not all cases, the accuracy of descriptions of runtime behaviour may be checked by relatively simple, automatic conformance testing procedures, which might be built into a future test harness and integrated with a future packaging tool.
- Where errors in declarations of runtime behaviour cannot be identified before deployment, there should be mechanisms built into future versions of the profile and its associated infrastructure which allow for the automatic reporting of errors to a certifying authority, which can manage a process for corrective action.
- In the meantime, the VLE may need to take remedial action, either by correcting its copy of any descriptions of runtime behaviour to coincide with the runtime behaviour which the object displays in practice; or by ignoring runtime behaviour which was inconsistent with the declared descriptions. Where reported marks exceed the declared absolute maximum, it would be permissible to reset the reported marks so that they equalled the absolute maximum.

Declaration of ability to save state

Just as the ***scoreReturned*** metadata field is needed to declare the total marks which may be awarded by a SCO, so there would be benefit in requiring SCOs to declare whether they are capable of saving state at runtime. This will in future allow a VLE to manage the saving of state, for example by giving the teacher the option to specify whether the saving of state should in some circumstances be turned off.

A third extension to the metadata attached to a resource node of ***technicalResourceType="interactive"*** should be ***canSaveState***, with possible values of ***"true"*** or ***"false"***.

`

Tracking time

The tracking of time spent on a task using the runtime is simple to implement and would be useful in helping to refine the accuracy of ***typicalLearningTime*** values by the collation of usage data (see page 35).

Although the VLE can track time elapsed between the launch of a resource and its termination, a measure of this time as measured by a SCO itself will be more accurate, as the SCO will be able to discount down-time, such as will occur during the loading of the learning object or if the SCO is paused by the student.

The learning object is responsible for reporting the time spent during the current session, which is reported to the VLE in *session_time*. The VLE is then responsible for aggregating the time spent in consecutive sessions into *total_time*, which can be passed back to a SCO which was resuming a suspended attempt and which wished to display this data or modify its behaviour accordingly.

Identifying the student

It is proposed that the VLE should identify students in two different ways to third-party content:

- by a form of address;
- by a unique but anonymous GUID.

For a student, the form of address should be a nickname, if the VLE has access to one, or alternatively a first name; for a teacher, the form of address should be a concatenation of title and surname, as in “Dr Jones”.

It is proposed that the unique but anonymous identifier should be a GUID allocated by the VLE.

There are four issues associated with the allocation of the identifier.

Data protection

It is proposed as a general principle that the identify of a student should not be released to third-party content providers. There would, in effect, be an assumed “wall of confidentiality”, with the VLE inside the wall and content providers outside the wall.

All that a content provider would know is that a student called “Dave” with an ID of A38C517F-8B3F-4C50-B513-AD85C7ECF83C was the same “Dave” that accessed the content publisher’s site last week.

This principle requires two supplementary rules:

- that conformant VLEs should never distribute or display the student GUID at the same time as any other identifying student details, expect

`

in circumstances which will need in future to be provided to allow student details to be transferred from one trusted VLE to another;

- that content publishers, who are given the GUID, should never try to solicit from students any personal details other than their form of address.

Management of student data

Once a third party website has a unique identifier for a student, it can start to implement VLE-type management functions, including the profiling of student competency in customised ways and management of student preferences.

Any VLE functionality implemented on the publisher's website is likely to fall beyond the teacher's control and in some circumstances this is likely to be undesirable. Content publishers should probably not save state on their own site, as there are circumstances in which a classroom teacher may wish to delete state (for example, before revisiting an exercise which has been assigned as an end-of-unit assessment). Similarly, if competency profiling data can be standardised and shared with other learning objects through the VLE, its utility is likely to be much enhanced.

On the other hand, there seems to be little reason why student preference data should not be stored locally; and even with competency data, there may be much to be said for keeping a local, "native" copy of the data, and for pioneering new ways of measuring competency without having to go through lengthy standardisation processes first.

Multi-player gaming

The passing of a unique student identifier will also represent an important pre-requisite for multi-player gaming and collaborative activities.

A future version of the profile could allow the learning object to discover from the VLE, not only that it was being used by a student A38C517F-8B3F-4C50-B513-AD85C7ECF83C called "Dave", but that Dave was collaborating with student CDE775C2-58ED-11D6-BA74-80C077C10801 called "Jane" and that when Jane logged on, they should be put in touch with each other in whatever way the multi-player content supported. Dave and Jane would not necessarily need to access the third-party service from the same VLE.

Automatic authorisation

The unique but anonymous identifiers may also be used in any future scheme for automatic authorisation of learning objects.

`

Summary of recommended runtime data fields

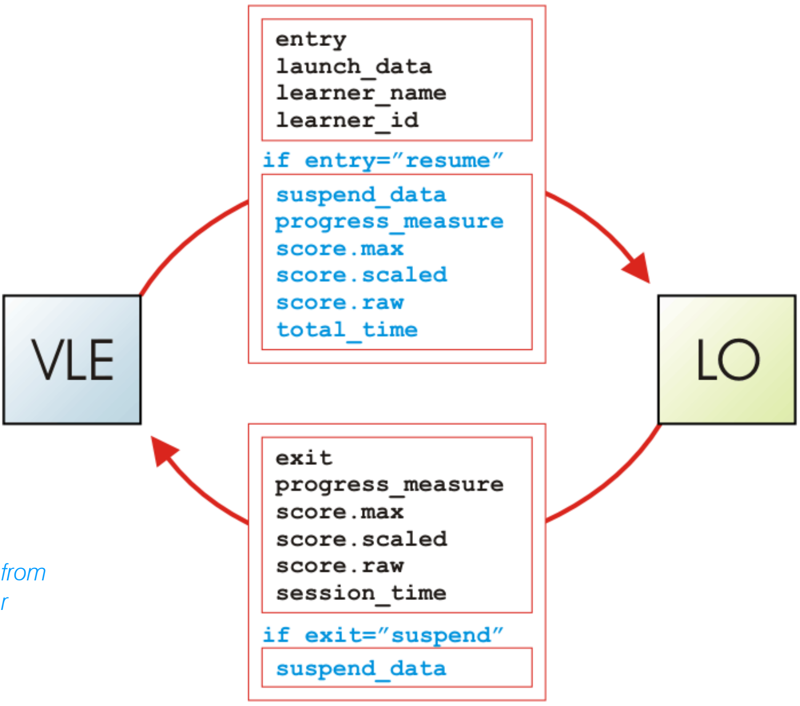


Figure 23. Summary of the fields from the SCORM runtime proposed for inclusion in the current profile.

Learning Object Metadata

Content publishers believe that any requirement for metadata should be kept simple. They are wary of the substantial overhead of tagging which was imposed on them by Curriculum Online.

In common with many standards-related difficulties, the use of metadata represents a chicken-and-egg impasse. VLE providers may see little point in building functionality on top of metadata fields which are not widely and consistently used; and publishers are not willing to use these fields unless they enable VLE functionality which enhances the utility of their content.

The rationale proposed for the requirements in this section is to to reduce the number of metadata fields to those where a clear case can be made for their use—but then to make a greater proportion of those fields mandatory to encourage the development of associated VLE functionality.

Publishers will remain concerned to minimize the cost associated with tagging and authors of UGC may be reluctant to tag at all. There is much that could be done to lighten this load by the development of appropriate tools which could manage the auto- or batch-completion of fields on the basis of user defaults or the type of authoring tool being used, and automatically upgrade legacy records.

`

Requirement for multiple metadata profiles

The *SALTIS Proposal* envisaged three metadata profiles, to be attached respectively to

- the manifest node;
- learning objects, however they are declared;
- individual supporting files.

In the light of the need to distinguish between assignable resources and non-assignable assets, it is proposed to create a fourth metadata profile to be applied to non-assignable assets.

Short and long titles and descriptions

The *SALTIS Proposal* envisaged the need for short and long titles and descriptions. LOM titles have a SPM[2] of 1000 characters, which is clearly far too long to be accommodated within a list (such as is illustrated in Figure 6) or on a badge (as illustrated in Figure 15). The longer the titles which need to be accommodated in the VLE-provided navigation pane, the less screen space will be available for the displaying of content.

An alternative, mandatory ***`title`*** field is included in the content package and this has an SPM of 200 characters. Even this is too long. It is therefore proposed that the SPM of the title field in the content package should be reduced to 50 characters, while the longer title of up to 1000 characters from the LOM can still be included in a special *About…* screen or dialog.

Similarly, there are two alternative fields for descriptions available in LOM: ***`general.description`*** and ***`educational.description`***. It is proposed that these should be used to provide a short description in ***`general.description`***, with an SPM of 200 characters; and a long description, with an SPM of 2000 characters in ***`educational.description`***. Both descriptions would be in plain text and targeted at teachers. The short description could be shown in the badge shown to teachers, while the long description could be shown in the *About…* display.

The SALTIS Proposal also suggested a new metadata field for student-targeted, formatted guidance. Although this would clearly be useful, the display of this guidance would complicate any student interface, and it is therefore not proposed to include this in the first version of the profile.

[2] Smallest Permitted Maximum: the length of title that a SCORM-compliant VLE must support before truncation.

`

LOM General section

It is proposed that there should be at least one unique *`identifier`* of type GUID, that *`title`* and *`description`* should be mandatory and *`keyword`* should be optional.

Unique identifiers

It is desirable, for the purpose of maintaining relationships between learning objects and supporting the future implementation of user reviews, that all learning objects should have a globally unique identifier.

LOM metadata can accommodate up to 10 identifiers, which may include, for example, ISBN numbers. But not all learning objects will be included on any one of these catalogues and so none of these alternatives meets the need for a common, universal and globally unique ID. The allocation of unique identifiers by a central authority introduces cost and complexity. It is therefore proposed that all compliant instances of the LOM should include a mandatory identifier which should be based on GUID (Globally Unique Identifier). GUIDs can be created by any personal computer with an extremely high probability that they will be unique.

LOM Lifecycle section

It is proposed that *`version`* should be mandatory along with at least one *`contribute`* record and, where *`copyrightAndOtherRestrictions="yes"`*, this mandatory *`contribute`* record should be of type *`rights holder`*.

Version information

It is likely that the management of lifecycle and versioning information will be an important requirement in future versions of the profile. Publishers will wish to update their content but this process may be problematic where teachers have created content aggregations which may be disrupted by updates to constituent learning objects. It is possible that an updated version of a SCO might return a different number of marks to an earlier version and that any automatic updating could break sequencing logic built on the earlier version.

The management of content lifecycle is therefore an issue what will need to be revisited in future versions of the profile. Looking forward to this future requirement, it is proposed that *`lifecycle.version`* should be mandatory, with a recommended default value of *`"1.0.0"`*.

Contributor records

It is proposed to adopt a vocabulary broadly consistent with the NDRB vocabulary for *`lifecycle.contribute.role`*:

- *`author`*
- *`contentProvider`*
- *`publisher`*
- *`rights holder`*

`

It is proposed to omit the NDRB value “UK rights holder”, replacing this with an optional and more flexible `territory` element:

```
<contribute>
    <role>
        <source>LOMv1.0</source>
        <value>rights holder</value>
        <ukscpp:territory>uk</ukscpp:territory>
    </role>
    <entity>…</entity>
    <date>…</date>
</contribute>
```

Guidance should be provided on the formatting of vCard records in the `lifecycle.contribute.entity` field.

LOM Meta-metadata section

As it is not anticipated that there should be any more than one authoritative LOM metadata record per learning object, there does not seem to be any requirement to complete most of the meta-metadata section, which will remain optional, with no obligation on the VLE to make use of any information provided in this section.

LOM Technical section

It is not proposed to include any fields from the technical section in the current version of the profile.

LOM Educational section

Much bewilderment has been expressed about fields such as `semanticDensity` and `interactityLevel`. It could be argued that `semanticDensity` is a useful concept given an agreed, automatic algorithm for calculating this figure and equating it to a typical reading age. But such an automated process could be completed by a VLE without any requirement for metadata tagging.

In spite of the general disfavour into which the section has fallen, it is nevertheless proposed that `learningResourceType` and `typicalLearningTime` should be mandatory on the grounds that, with consistent implementation, these fields will provide important information. It is proposed that `intendedEndUserRole`, `typicalAgeRange` and `education.description` should be optional.

`LearningResourceType`

Given consistent application and an agreed vocabulary, it is anticipated that this field may be found to be useful in supporting teacher search and informed purchase (see *Profiles, feature-grids and other declarable data* on page 45).

The difficulty with this field lies in the definition of a useful vocabulary. The UK LOM Core project (2004) notes that “Use of the LOMv1.0 vocabulary is problematic as it includes terms that describe both the form (e.g. diagram) and the function (e.g exam) of the object”. A good technical vocabulary has mutually exclusive terms (as in “Red”, “Blue”, “Green” but not as in “Red”, “Blue”, “Green”, “Light”, “Dark”). Non-mutually exclusive terms arise when the terms in a vocabulary describe more than one characteristic.

The CELEBRATE vocabulary shown in Appendix A, which was adopted by Curriculum Online, is superior to the IEEE/LOM vocabulary in that it generally attempts to describe only the form of the resource and not its purpose. In order to achieve clearly defined, mutually exclusive terms, it also tends to use fairly abstract terms: “exploration” covers simulations, experiments and role-play games for example. Such a level of abstraction has the disadvantage that it is often not always very teacher-friendly. Indeed, the distinction between form and function may not be uppermost in most teachers’ minds.

This point is illustrated by the fact that the CELEBRATE vocabulary has been recently extended by the National Digital Resource Bank (NDRB) to include terms such as *Project*, *Worksheet*, *Warmer or filler*. These terms doubtless represent a response to teacher demand but reintroduce the problems of the original IEEE/LOM vocabulary in describing function as well as form and therefore creating terms which are not mutually exclusive – it is likely, for example, that a *Warmer or filler* (function) might also be a *Drill and practice* (form).

The existence of terms which are not mutually exclusive can be mitigated by the ability under LOM to put a resource under more than one category; but this is only a partial solution. LOM specifies that the first term in the list is to be regarded as the “dominant” term, to be used when each resource needs to be allocated to a single category. The ability to make multiple descriptions should be used where resources genuinely combine multiple resource types, and should not be used as a fix for a poor vocabulary.

The challenge is to create a vocabulary which is technically sound while also using terms which are familiar and useful to teachers.

The approach recommended here is to adopt a hierarchical vocabulary. At a higher level are terms which will tend to be abstract but which are carefully selected to be mutually exclusive. Each of those abstract terms can then be sub-divided into more concrete, familiar terms. The NDRB term *Simulation* then becomes a type of *Exploration*, as defined by CELEBRATE; *Warmer or filler* becomes a type of *Presentation aid*, making it clear that the term cannot normally apply to something which might be described as *Drill and practice*.

One advantage of a hierarchical vocabulary is that it allows for graceful degradation. A system which does not recognise the NDRB term *Simulation* can properly map anything tagged as a *Simulation* to the higher-level term *Exploration*.

Once the principle of a hierarchical vocabulary is accepted, the CELEBRATE vocabulary can itself be further rationalised: *Drill and practice*, for example, is clearly a type of formative *Assessment*.

The NDRB vocabulary does highlight some gaps in the CELEBRATE vocabulary, which generally appears to be applicable to atomic resources assignable to individual students. There are no terms appropriate to aggregations (Course, Lesson plan etc.), assets, or resources intended for use in a whole-class setting, for example on a whiteboard.

What is therefore proposed is a three-level hierarchy, the top level corresponding to different types of object shown in Figure 2, which have been codified as terms in the `technicalResourceType` vocabulary: asset, learning object, and aggregation. As each of these three types of object will have a different LOM profile, the three different branches in the vocabulary could be seen as representing three different vocabularies.

It is not proposed to create any nested vocabulary within the profile for the *`technicalResourceType “Asset”`*, as assets will in practice be distinguished by their file type.

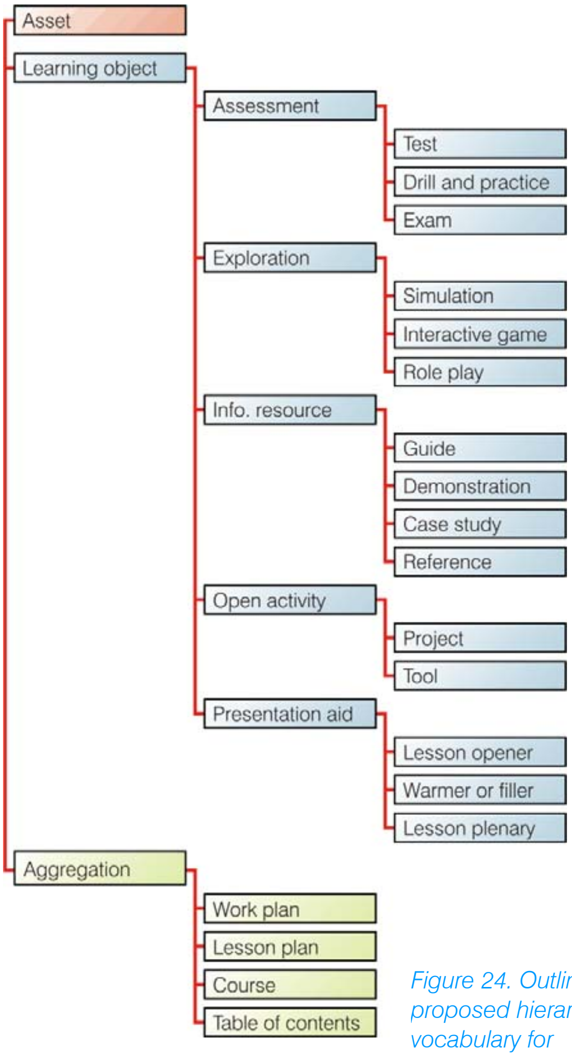


Figure 24. Outline of proposed hierarchical vocabulary for learningResourceType.

The vocabulary under *`technicalResourceType “Aggregation”`* is at this stage only indicative: the only one of these terms which it is anticipated will be supported in the first version profile is *`“Table of contents”`*, which will be the recommended interpretation for a vanilla IMS *`organization`*.

The detail of this proposal will need to continue to be developed through further consultations with publishers, VLE suppliers and ecosystem suppliers.

SCORM suggests an SPM of 10, meaning that a resource could be given 10 different values for *`learningResourceType`*; but this value appears excessive. In the interests of avoiding over-tagging, it is proposed that the SPM should be reduced to an SPM of 3.

The “assumed” term *`Asset`* is proposed to describe a resource which it would be meaningless to assign to the student but which might be used by a

`

teacher or student in the creation of another resource or in the performance of another task. Image clip-art is an obvious example. The term is “assumed” because the nature of the proposed object is so materially different from other assignable resources that it is arguable that it is not a learning resource at all. It would certainly not make sense to assign a *typicialLearningTime* to a JPEG; and in general its metadata requirements will be lighter than a learning resource. For these reasons, it is proposed that asset should:

- be added to the proposed vocabulary for *resource.technicalResourceType*;
- be provided with a separate and lighter LOM metadata profile.

IntendedEndUserRole

It is proposed to make *intendedEndUserRole* optional for both learning objects and supporting files, on the basis that where this value is not specified, the VLE will assume that *intendedEndUserRole=“Learner”*. The proposed vocabulary uses an elision of the *Celebrate* and *Curriculum Online* vocabularies:

- *Learner*
- *Teacher*
- *Manager*
- *Parent* (Curriculum Online only)
- *Author* (Celebrate only).

The value of *Author* lies in the possible requirement to include materials which provide guidance to support the re-use and re-aggregation of content.

VLE providers should have a basic concept of user role to ensure that objects which are declared to have an *intendedEndUserRole* are only shown to users within that role.

TypicalAgeRange

References to age may have three different meanings:

- chronological age, with implications for a student’s attitudes and interests;
- the age at which a student normally studies particular topics within a particular curriculum (key stage),
- developmental age (as in reading age).

TypicalAgeRange refers to the first of these: the last two are covered by vocabularies which are available for use within the classification section and the use of which it is proposed should be mandatory.

For most students, the three types of age will be roughly aligned and for this reason, *typicalAgeRange* is optional. Where it is not supplied, the VLE can assume that the age range at which a resource is targeted will be indicated by any key stage data included within the classifications section.

`

The `typicalAgeRange` element should therefore only be used when the intended student's chronological age differs from the normal age range associated with the appropriate key stage data (as, for example, in a resource designed to teach basic literacy to adults). Otherwise, VLEs should be free to assume an implicit `typicalAgeRange`, deriving this value from keystage-related classification data.

`TypicalLearningTime`

The disadvantage of this field is that any judgement of typical learning time is likely to be subjective and there is no well-established method for its determination. Publishers may be inclined to over-rate the learning time for their own materials.

Balancing the undoubted difficulties of this field is its considerable utility, both in informing purchase (see page 45) and in enabling teachers to plan lessons. There would be considerable value in getting this field right—and to this end, the following suggestions are made:

- Guidelines should be drawn up which would help standardise the determination of `typicalLearningTime` for standard content types, such as text and simple QTI-type interactions.
- Publishers with non-standard tools should create their own guidelines through practical trials.
- In the future, suggested values could be calculated by a tagging tool, on the basis of the guidelines.
- While the LOM does not allow the provision of a range of values, the project guidelines could suggest a suitable policy on rounding values to provide an appropriate level of approximation.
- In the future, values should be validated and, if necessary, updated, by real data returned from conformant VLEs to a central repository.

With all these considerations in mind (including the argument on page 45), it is proposed that `typicalLearningTime` should be mandatory for all learning resources except assets, for which the concept of `typicalLearningTime` is inapplicable.

Description

It is proposed that `educational.description` should be used for the long descriptions described in *Short and long titles and descriptions* on page 29. This field is recommended as optional, on the grounds that, if not provided, the mandatory short description provided in `general.description` can used in its place.

LOM Rights section

It is proposed that `rights.cost` and `rights.copyrightAndOtherRestrictions` should be mandatory and that `rights.description` should be optional.

`

Following an original recommendation made in *The SALTIS Proposal*, If `rights.copyrightAndOtherRestrictions="yes"` then the VLE should show a short copyright notice within the badge display (see page 13) along the following lines of: "Copyright © <year> <holder>" where <year> is derived from the date(s) attached to `lifecycle.contributor.date` element(s) and <holder> is taken from the `contributor.rights holder` field (see *Contributor records* on page 30). If the `rights.description` is completed, then the short copyright notice should link to the long copyright notice displayed in the `About`… display.

If `copyrightAndOtherRestrictions` does not equal `"true"`, then the short copyright notice should read "Published by…", "Written by…" or "Supplied by…", depending on the value of the `role` element within the first contributor element supplied.

LOM Relations section

Product, branding and thumbnail records

It is proposed that the profile should support branding of learning objects by icons and, less prominently, supplier logos. Both of these images should be supplied in a range of different sizes, like desktop icons, so that they can be used flexibly by the VLE in different circumstances. To avoid duplication of data and to allow for updating, these image files should stored remotely and referenced from the content package. It is also proposed to include screenshot or preview images at a single size, on the assumption that these can be sized and used in the place of icon images within the teacher interface, as required. Icon and preview images are referred to collectively as thumbnails.

One way of satisfying this requirement would be for the specification of icon and branding records to be located either on the publisher's website or on a common repository. Suggestions for the formatting of these records are contained in the *SALTIS Proposal*.

Learning objects could associate themselves with a particular product record, branding, icons and preview, by using the relations section of the LOM.

It may be that the creation of external metadata links could be handled using semantic web technologies being developed as part of the ISO/IEC Metadata for Learning Resource (MLR) initiative. Such a technology could in the future also provide a basis for further extensions of the LOM metadata to support folksonomies.

It is proposed that the support for thumbnails of both kinds should be made mandatory, for three particular reasons:

- The implementation of thumbnails will have a significant impact on the "look and feel" of VLEs – and an inconsistent implementation will have a correspondingly detrimental effect to the user interfaces of VLEs.
- Thumbnails are likely to be one of the most immediately visible attributes of conformant content. Consistency in this requirement will

`

help create brand value for the profile, which will in turn help drive demand for conformant content.

- Publishers gave strong generally strong support for the principle of thumbnails and it seems difficult to anticipate any publisher would not want to enhance the visibility of content within VLEs.

Vocabulary for ***relation.kind.value***

Term	Catalog type	Comment
isPartOfProduct	URI	To link to a product record
hasBranding	URI	To link to a branding record
hasIcon	URI	To link to an icon record
hasPreview	URI	To link to a preview graphic
assesses	GUID	References another learning object
isAssessedBy	GUID	References another learning object

There may be other suggestions for this relations vocabulary.

IEEE terms such as *isPartOf* and *requires* have not been included on the grounds that any such relationships will be evident from the structure of a content package and therefore need not be repeated in the metadata.

While publishers may wish to provide "See also…" relationships, these may also be implicit within the structure of a package. There is a danger of the pedagogical value of the relationship being devalued by excessive provision of commercially-motivated links and it is likely that this kind of data would be of more value if it were derived, as it is on Amazon, from usage.

Nor have commercial relations such as *hasPicture*, *hasDemo* have been suggested, as an no e-commerce features are being included in the first version of the profile.

In the absence of a central index provided by a national ecosystem, the ability of the VLE to resolve links to other learning objects will be dependent on those learning objects having already been loaded into the same VLE.

LOM Annotation section

It is proposed that the annotation section should not be supported.

LOM Classifications section

It is proposed that use of the classifications section should be made mandatory at a simplified level.

`

A broad approach to this section is to divide vocabularies into three categories:

- High level curriculum vocabularies (e.g. subject & key-stage);
- Vocabularies describing educational-level;
- Detailed curriculum vocabularies (e.g. QCA programmes of study).

It is proposed that it should be mandatory for content publishers to include at least one classification from the first two of these categories, where an appropriate vocabulary from the following lists exists. The use of vocabularies from the third section would be optional.

Appropriate high-level curriculum vocabularies available on Vocabulary Bank are:

- COL – Subject Keystage View
- COL – MFL Language Selection (there currently appears to be a problem with this vocabulary's placement of key stage information);
- COL – Special Educational Needs
- LTS – Early years curriculum
- LTS – National qualifications
- QCA – Cross curricular skills.

Appropriate vocabularies for educational-level are:

- COL – Attainment level.

All other vocabularies currently on Vocabulary Bank would fall into the third, optional category.

There has been an interesting attempt to create a central mapping for educational level (see http://www.ukoln.ac.uk/metadata/education/ukel/) and there might be scope for carrying this approach forwards, perhaps in the context of a new version of Vocabulary Bank.

There are significant gaps for both curriculum and educational-level vocabularies outside the national curriculum. The need to maintain, manage and extend the vocabularies presented on Bank may be an issue which will be addressed in the third part of the project.

Suggested metadata profiles

The *SALTIS Proposal* was for the creation of three separate metadata profiles for metadata records to be attached respectively to:

- the manifest node;
- individual learning objects or resources (it is a matter for the technical team to determine how these are declared, depending on the way in which the profile implements the disaggregation model);
- supporting files, which may be accessed in the context of a particular learning object.

`

Two further profiles are now proposed:

- aggregations
- non-assignable assets.

It is proposed that the profile for aggregation would be identical to the profile for resources, except that the aggregations would have a different vocabulary for `learningResourceType`.

Centralisation of metadata

There is an implicit containment hierarchy, shown in Figure 25, between the five different profiles. Resources are not really "contained" within aggregations but rather referenced by them.

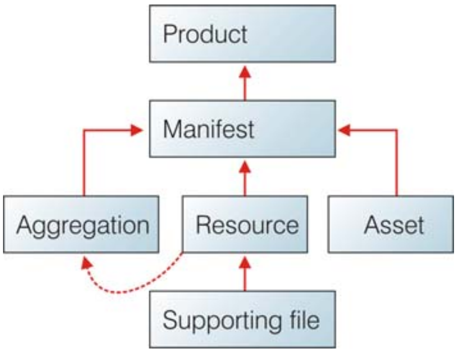


Figure 25. Implicit containment hierarchy.

At the top of the containment hierarchy is a product record, which probably needs to be completely separate from the content package, and might contain the following data:

Field	M/O	SPM	Description
Identifier	M		GUID
Version	M		e.g. "1.0.0"
Title	M	50	Name of the product
Description	M	200	
Page	O		Link to a product page
License	O	2000	Text of a click-through license to which the purchaser needs to agree before use
Logo	O		URL of an XML record referencing a multi-resolution logo.
Icon16	O		URL of a GIF image which can be used as an icon for the product at various resolutions.
Icon 32	O		
Icon 48	O		
Icon 64	O		

The containment relationships between supporting files and resources, and between resources, aggregations and assets and manifests are all declared implicitly by the structure of a content packaging file; but the containment of a manifest within a particular product must be declared explicitly by an `isPartOfProduct` entry in the manifest's metadata.

Unnecessary repetition of metadata can be avoided by placing data in a single place and then referencing that data from multiple places. Such an approach

`

reduces file sizes (particularly when the size of the data is significantly greater than the size of the reference or pointer, which must still be included multiple times). Even more importantly, centralisation of data facilitates maintenance, as only a single copy of the data need be updated.

There may well be scope for metadata to be centralised, either by moving it to the manifest node, or by moving it onto records (like the product record shown above) which can be hosted remotely on the publisher's website and referenced as required.

A potential disadvantage of centralised metadata lies in the possibility that a resource might become separated from its manifest.

This possibility also applies to the original metadata which, in the wrong hands, may be separated from the resource itself. The only reliable way of avoiding any of these risks is to implement a method of automatic authorisation which allows ultimate control of launch to remain with the content publisher and ensures that launch is managed only by trusted VLEs.

In the meantime, it is anticipated that any future regime to support the redistribution of resources would require content to be redistributed within its original manifest wrapper (see last paragraph of *A terminology for* aggregation and organisation

The following terminology is proposed to help clarify the discussion of aggregation.

Term	Meaning
Learning content	Abstract term for all the digital "stuff" that can be managed by a VLE and which delivers learning. This will include static data (such as text and video) as well as applications, plug-ins, players and widgets.
Resource (technical)	An atomic piece of learning content. Although a resource may often be composed of many files, those files should be regarded (either for technical or commercial reasons) to be inextricably bound together.
Resource (non-technical)	An abstract term to describe assets, learning objects, and aggregations (see below).
Asset	A resource which is not intended to be assigned, but rather used freely in the creation of other learning objects and the completion of tasks. Clip-art is a common example.
Learning object	An individual resource which may be assigned for the purposes of learning. The term "object" blurs the distinction between a data file and the application that manipulates it—"objects" may therefore include applications, services and widgets as well as static data. From the perspective of any further aggregation, it is the learning object which

	should be regarded as the basic atomic unit.
Aggregation	A collection of learning objects (and other aggregations) which can be made the subject of a single assignment.
Interstitial data	All data within an aggregation which is not defined as being part of asset or learning object. Interstitial data cannot be reused in other contexts but is bound to a particular aggregation. It can be thought of as the mortar which binds together the reusable bricks in a particular wall.
Orchestration	The process of managing the delivery of an aggregation.
Assignable unit	An atomic object or aggregation that is intended to be made the subject of a single assignment.
Assignment	The allocation of an assignable unit to a particular assignee (see page 15).
Organisation	The management of assignable units or the structure (such as a hierarchy of nested folders) within which these assignable units are organised. Note that within this proposal, the ***`organization`*** node in an IMS content package should be regarded as the root node of an aggregation and **not** an organisation.
Folder	A container within an organisational structure. Following the observation above, the folder will in future be represented within a content package by an IMS ***`manifest`*** node and not by an ***`item`*** node under an ***`organization`*** node.
Sequencing	A form of aggregation in which objects are launched one after another. There are two fundamental types of sequence: *flow* sequencing, in which the order of objects is determined automatically by the VLE; and *choice* sequencing, in which the basic navigational paradigm is one in which the student chooses what to do next. Note that a choice sequence may in practice appear to be very similar to accessing a folder in the VLE's organisational structure: the difference is that, where such a concept is applicable, the choice sequence is treated as a single assignment while the organisational folder is treated as a collection of different assignments.

Figure 1. Glossary of terms relating to aggregation.

Packaging is ephemeral on page 2). With this in mind, the profile should consider the centralisation of metadata where possible.

Key to the metadata recommended profile grids

M	A mandatory section
O	An optional section
N	A section which is not supported
M (1)	A mandatory element which must be supported both by VLEs and publishers (also showing, where more than one record is allowed, the minimum number of instances to be included).
O	An optional element from the point of view of content packagers. From the perspective of the VLE, all optional elements must be supported appropriately.
m/o	A conditional element, which is dependent on whether an optional container is provided
N	An element which is not supported in the profile.
SPM	Smallest permitted maximum – for example, the number of characters that must as a minimum be allowed by the VLE before truncating a string.

Where not specially mentioned, values for SPMs should follow recommendations in SCORM 2004 3rd Edition.

	Manifest		Resource		Asset		Supporting File		Notes
	?	Format	?	Format	?	Format	?	Format	
General	M		M		M		M		
Identifier (of the resource)	M(1)		M(1)		M(1)		N		
Catalog type	M	URI, DOI, ISBN, Guid	M	URI, DOI, ISBN, Guid	M	URI, DOI, ISBN, Guid	N		Guids may be generated automatically by the packaging/tagging tool. Further identifiers can be added by the user who wishes to use other catalog types.
Catalog entry	M		M		M		N		
Title	M	SPM 50	M	SPM 1000	M	SPM 50	M	SPM 50	For LOs, this is the place for a long title, the title within the content package having SPM 50.
Language (of resource)	N		O		O		O		If not present, language is assumed to be English.
Description	M	SPM 200	M	SPM 200	O	SPM200	N		This is the place for the short description. A long description, if required, can be included in the Education section.
Keyword	N		O		O		N		Separate uses of the keyword element should relate to the provision of keywords in different languages.
Coverage	N		N		N		N		Use *General.Keyword* instead
Structure	N		N		N		N		
AggregationLevel	N		N		N		N		
Lifecycle	M		M		N		N		
Version	N		M		N		N		Use “0”..”9” & “.” only, with the format “<Major>.<Minor>.<Revision>” e.g. “1.2.34567”. The packaging tool should increment <revision> automatically.
Status	N		N		N		N		
Contribute	M(1)		M(1)		N		N		
Role	M		M		N		N		
Entity	M		M		N		N		
Date	M		M		N		N		

Figure 26. Suggested profiles of LOM General and Lifecycle sections

	Manifest		Resource		Asset		Supporting File		Notes
	?	Format	?	Format	?	Format	?	Format	
Meta-metadata	M		M		M		M		
Identifier (of the metadata)	N		N		N		N		
Catalog	N		N		N		N		
Entry	N		N		N		N		
Contribute	N		N		N		N		
Role	N		N		N		N		
Entity	N		N		N		N		
Date	N		N		N		N		
MetadataSchema	M(2)	LOMv1.0 SMP-Mv1.0	M(2)	LOMv1.0 SMP-Rv1.0	M(2)	LOMv1.0 SMP-Av1.0	M(2)	LOMv1.0 SMP-Fv1.0	Used to identify which of the four profiles the metadata record is using. “SMP” used as an interim name.
Language (of metadata)	N		N		N		N		
Technical	N		N		N		N		
Format	N		N		N		N		
Size	N		N		N		N		
Location	N		N		N		N		
Requirement	N		N		N		N		
Type	N		N		N		N		
Name	N		N		N		N		
MinimumVersion	N		N		N		N		
MaximumVersion	N		N		N		N		
InstallationRemarks	N		N		N		N		
OtherPlatformRequirements	N		N		N		N		
Duration	N		N		N		N		

Figure 27. Suggested profiles of LOM meta-metadata and technical sections

	Manifest		Resource		Asset		Supporting File		Notes
	?	Format	?	Format	?	Format	?	Format	
Educational	O		M		O		M		
InteractivityType	N		N		N		N		
LearningResourceType	N		M	SPM 3	N	Always “asset”	M	SPM 3	
InteractivityLevel	N		N		N		N		
SemanticDensity	N		N		N		N		
IntendedEndUserRole	N		O		O		O		
Context	N		N		N		N		
TypicalAgeRange	N		O		N		N		“0”..”9” & “-“only, showing years (single number or range). Examples: “8” or “10-12”.
Difficulty	N		N		N		N		
TypicalLearningTime	N		M		N		N		
Description	O	2000X10L	O	2000X10L	N		N		This is the place for a long description, if required
Language (of intended user)	O		O		O		O		
Rights	M		M		M		N		
Cost	M		M		M		N		See Contributor records on page 30.
CopyrightAndOtherRestrictions	M		M		M		N		
Description	O		O		O		N		
Relation	O		O		O		N		
Relation	O		M(2)		O		N		
Kind	m	isPartOfProduct	M	hasIcon	m	hasPreview	N		
			M	hasPreview	N		N		
				assesses					
				assessedBy					
Resource	m		M		N		N		
Description	N		N		N		N		

Figure 28. Suggested profiles of LOM Educational, Rights and Relation sections

	Manifest		Resource		Asset		Supporting File		Notes
	?	Format	?	Format	?	Format	?	Format	
Annotation	N		N		N		N		
Person	N		N		N		N		It is intended that a future SALTIS standard will include a profile dedicated to annotations / reviews.
Date	N		N		N		N		
Description	N		N		N		N		
Classification	N		O		N		N		
Purpose	N		N		N		N		Deprecated for reasons given elsewhere by Schemeta.
TaxonPath	N		O		O		N		See “Classification vocabularies”
Source	N		m		m		N		
Taxon	N		m		m		N		
Id	N		m		m		N		
Entry	N		m		m		N		
Description	N		N		N		N		
Keyword	N		N		N		N		Use *General.Keyword* instead

Figure 29. Suggested profile of LOM Annotation and Classification sections

Profiles, feature-grids and other declarable data

While it is not within the scope of this stage of the project to consider the need for conformance testing and kite-marks, it is worth at least considering the likelihood that any profile, to be effective in the marketplace, will need to be supported in this way.

A key criteria of any kitemark is that it should provide confidence to end-users that products which say they will interoperate should do so in practice and in ways which conform to user expectations.

The desire to create a kitemark which provides a robust guarantee of interoperability may, in certain circumstances, tend to restrict the diversity of products. This management of conformity while enabling diversity is critical to the project's objectives.

In some circumstances, the solution may be to create more than one kite-mark but in most circumstances it will be desirable to limit the excessive proliferation of alternative kitemarks. A single kitemark may cover a number of related profiles by appending a feature grid, which may also carry supplementary, objective information designed to support informed purchasing decisions.

While it is not within the scope of this project to determine what such a feature-grid may look like, it would be useful to consider the metadata profiles suggested above in relation to their role in providing data to support such a feature grid in the future.

It is suggested that the main purposes of a feature grid for the content packaging profile to be produced by this project are to allow the buyer to see:

- the extent to which a product is capable of disaggregation;
- a measure of the estimated total learning time provided by these objects;
- the types of learning object provided;
- the extent of functional integration with the learning objects provided and a conformant VLE;
- the top-level curriculum areas (subject/keystage) and educational level and anticipated age range that the content addresses.

The second point in this list is critical. Any feature grid which showed numbers of learning objects without showing their ***`typicalLearningTime`*** would create a distorted impression of the value of products containing large numbers of small objects. It is principally to create a balanced feature-grid that ***`typicalLearningTime`*** is being proposed as mandatory.

Figure 30 shows what such a learning grid might look like. All of this data could be automatically derived from the metadata profiled in Figure 26–Figure 29. It is proposed that kitemarked packages would, through the operation of approved packaging tools, automatically upload this data to a central certificating agency's repository, where the values for `typicalLearningTime` could be refined on the basis of data automatically submitted by conformant VLEs.

			Functions		Pedagogy			
Resource type	Total number	Total learning time	Report scores	Save state	Information	Guide	Exploration	Drill & practice
Assets	200	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Expositive resources	60	10h	N/A	N/A	10	50	-	-
Interactive resources	10	5h	9	10	-	-	1	9
Curriculum focus	Key stage 4 History Key stage 4 Art and design.							
Educational level	NC Attainment levels 5 – 8.							
Intended age range	14+							

Figure 30. Example of a possible feature grid which might be associated with certificated packaged content.

For the purposes of the function grid, resources of `technicalResourceType` `dynamic` and `static` are described under the single term "expositive resource", as the distinction between the two sub-terms is likely to be of limited interest to teachers.

A fourth resource type, *Tool*, is likely to be added in a second version profile, when the ability of VLEs to manage file associations and to edit initialization data will permit the integration of fully reusable tools. In the first versions of the profile, it will be possible to embed tools within the interactive resources but not to provide these tools to the user without an associated learning context.

Appendix A – CELEBRATE terms for *`learningResourceType`*

Assessment
Assessment and evaluation items. Exams and tests. Any LO whose primary purpose is the evaluation of the user's actions or input or to support teacher design or development of such materials. Used e.g. for assess learner performance or self-assessment.

Drill and practice
Simple exercises and games. Exercises (drills) that perform skill training are very condition and action specific. They usually contain only simple IF-THEN logic rules. Many ‘educational’ games belong to this category if they concentrate on specific skills.

Exploration
Simulations and experiments. Simulations are imitating the behaviour of some situation or by means of a suitably analogous situation or apparatus. Experiments are actions or operations undertaken in order to discover something unknown, to test a hypothesis, or establish or illustrate some known truth.

Glossary
Dictionaries and vocabularies. Collection of specialized terms and their meanings usually arranged in some stated order.

Guide
Manuals and tutorials. Manuals provide guidance on the particular topic (e.g. roadmap, hints, etc.) and are usually also intended to be kept at hand for reference. Tutorials are resources that provide guided, practical information about a specific subject.

Information resource
Pictures, texts, videos, presentations, collections and databases. Any presentation or informative content that is ‘raw’ material for learning.

Open activity
Artistic projects and creative exercises. Projects and exercises that are not very confined or limited. Many more complicated games that require more than simple logic belong to this category.

Tool
Editors and other kind of programs for producing something. Editors can process e.g. text or pictures and they can be used for creating and editing other LOs. Tools can also perform calculations or conversions.

Extract from Celebrate Application Profile, published November 17 2003, available at http://celebrate.eun.org/docs/CELEB_AP_v1.1_2003-11-17.pdf.

Appendix B – mappings for *learningResourceType*

<table>
<tr><th>Celebrate/Curriculum Online</th><th>NDRB</th></tr>
<tr><td><i>Assessment</i></td><td><i>Assessment</i></td></tr>
<tr><td><i>Drill and Practice</i></td><td><i>Drill and practice</i></td></tr>
<tr><td><i>Exploration</i></td><td><i>Interactive game</i>
<i>Role play</i>
<i>Simulation</i></td></tr>
<tr><td><i>Glossary</i></td><td><i>Reference material</i></td></tr>
<tr><td><i>Guide</i></td><td><i>Worksheet</i></td></tr>
<tr><td><i>Information resource</i></td><td><i>Case study</i>
<i>Demonstration</i>
<i>Presentation</i></td></tr>
<tr><td><i>Open activity</i></td><td><i>Project</i>
<i>Warmer or filler</i></td></tr>
<tr><td><i>Tool</i></td><td></td></tr>
<tr><td><i>[Asset]</i></td><td></td></tr>
<tr><td colspan="2">The following NDRB terms both refer to aggregations. There is probably a need for such a vocabulary but it is probably too early to say how the tagging and description of aggregations should be handled.</td></tr>
<tr><td></td><td><i>Course</i>
<i>Lesson plan</i></td></tr>
</table>