PACKAGING AND PUBLISHING LEARNING OBJECTS: BEST PRACTICE GUIDELINES

January 2005

Copyright protection notice

Contents

1. Foreword

For developers of e-learning content, the education sector presents both a challenge and a tremendous opportunity. Education in the UK is going through a revolutionary change in the way that people teach and learn, with the emphasis on participation, widening access and personalisation. E-learning is an important driver in this transformation, and providers of content have a vital role to play in helping to shape this future.

Becta not only seeks to inspire industry to rise to this challenge, to forge new standards of excellence, but also to provide structure and cohesion through taking a partnering approach. For those already involved in providing content to education, and for those who value these new opportunities, Becta is a source of support and advice, a ready partner who is willing and able to help you contribute to delivering educational excellence for all.

2. Becta: Our role and vision

Becta is the Government's key partner in the strategic development and delivery of its information and communications technology (ICT) and e-learning strategy for schools and the learning and skills sector. Becta is a UK agency and we support all four UK education departments in their strategic ICT developments. We facilitate knowledge transfer among the departments in order to encourage innovation and improvement, and bring coherence and synergy to UK-wide developments.

Partnerships are central to our role and we will work across government and its agencies, with industry and other key stakeholders to provide support and choice for the locality, and with the locality to enable its work to meet national frameworks and standards. We will develop key strategic relationships with local and multinational ICT business and industry to best meet the needs of education.

Becta's mission is to exploit the power of ICT to support learning. We will provide strategic leadership on ICT and learning, helping to develop a world-class education system. Becta will guide and co-ordinate the necessary changes in policy and practice and broker effective partnerships to establish and exploit reliable and sustainable educational technology.

The learner is at the centre of our vision – learners of all ages and in all contexts: formal and informal learning within educational institutions, the workplace, the community, and the home.

This set of best practice guidelines has been developed in consultation with representatives of the e-learning industry to provide a framework for the achievement of learning content that is accessible, durable, reusable and interoperable across learning platforms used in education.

3. About these guidelines

The publication of this document is one of several interrelated initiatives undertaken by Becta, including the Becta Learning Platform Conformance Regime. These guidelines, together with the Conformance Regime, provide a standardised and agreed framework for the packaging of learning content, and its delivery and use in education. With an increasing number of different learning platforms now in use in compulsory education, the wide adoption of a standard, consistent and technically robust method of exchanging learning content packages between systems is essential to the future growth and success of ICT in education.

In January 2003, Curriculum Online was launched. Curriculum Online is central to the Government's drive to transform teaching and learning in schools by improving access to ICT and multimedia resources for all pupils. Its aim is to bring teaching professionals and multimedia resources together.

To help bring about this aim, the Government has set aside substantial funds in the form of eLCs (electronic learning credits). This eLC money goes straight to the school to spend on multimedia resources.

Curriculum Online offers easy, pinpoint searching through all of the thousands of available multimedia resources which can be bought with eLCs, as well as resources that are free. All these resources are from approved suppliers, and are aligned with the curriculum of subjects taught in schools in England.

Since the launch of Curriculum Online, many lessons have been learned. These guidelines are very much built on the lessons, experiences and insights gained through the work of the e-learning industry and key organisations involved in e-learning, such as the National Learning Network (NLN), the Centre for Educational Technology Interoperability Standards (CETIS) and the Joint Information Systems Committee (JISC).

The guidelines are aimed at developers who are already involved in the provision of digital learning content, as well as those who would like to get involved. They are not exhaustive by any means, but they do represent a practical guide to help ensure that your content is accessible and interoperable across learning platforms used in education. It is our intention to ensure that these guidelines remain relevant and valuable to content developers by evolving them and updating them in synergy with developments in technology and practice.

4. Packaging and publishing learning objects: An overview

Simply put, the process of packaging and publishing learning objects is the means by which collections of digital learning content can be assembled so that they can be found, accessed, used by learners, tracked and monitored through learning platforms. It is used to:

- ensure that the learning content runs within a learning platform environment as the developer had intended
- describe how the content is aggregated or structured and so determine how the materials will be presented to the teacher or learner
- describe how, if at all, the content can be disaggregated or broken apart into constituent elements so that these may be used, for instance, in different contexts.

The Metadata that forms part of the package is used to:

- store descriptive words that are relevant for people using free text search
- store information about the developer/publisher
- store information on the product's IPR status, and any restrictions on use
- map learning content to curriculum topics. (Note that for use in England, learning objects must be classified using the Curriculum Online Metadata Scheme.)

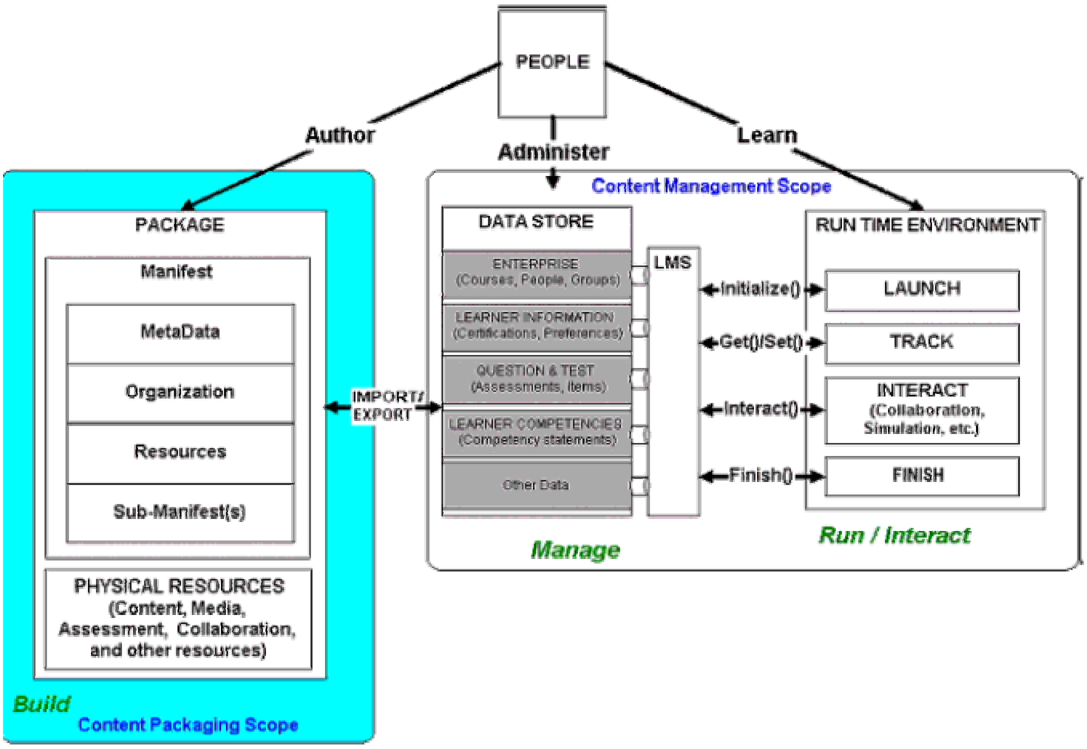
These guidelines explore all of these and more.

The Becta Learning Platform Conformance Regime[1] sets out the minimum technical and functional standards and specifications for learning platforms used in education. The guidelines contained here are consistent with this regime. It states that *conformant* learning platforms must be able to accept packages (that is, packaged learning content) that are *compliant* with the IMS content packaging data model and binding specifications (version 1.1.3) and the ADL Extensions. These specifications are embodied in the latest release of SCORM™ 2004 and are referenced throughout this set of guidelines. It should also be noted that version 1.1.4 of the IMS content packaging data model and binding specifications were published in November 2004. These guidelines will be regularly updated in accordance with other developments, and changes in the Conformance Regime.

The following diagram is reproduced from *IMS Content Packaging Best Practice Guide Version 1.1.3 Final Specification*. It illustrates the Content Framework, and describes the relationships between the three principal elements, namely, the Content Package, the Learning Platform and the Learning Run-time Environment.

[1] Becta Learning Platform Conformance Regime Version 2.3, September 2004

Diagram 1: IMS content specification


PEOPLE
Author
Administer
Learn
Content Management Scope
PACKAGE
Manifest
MetaData
Organization
Resources
Sub-Manifest(s)
PHYSICAL RESOURCES (Content, Media, Assessment, Collaboration, and other resources)
Build
Content Packaging Scope
IMPORT/ EXPORT
DATA STORE
ENTERPRISE (Courses, People, Groups)
LEARNER INFORMATION (Certifications, Preferences)
QUESTION & TEST (Assessments, Items)
LEARNER COMPETENCIES (Competency statements)
Other Data
LMS
Initialize()
Get()/Set()
Interact()
Finish()
RUN TIME ENVIRONMENT
LAUNCH
TRACK
INTERACT (Collaboration, Simulation, etc.)
FINISH
Manage
Run / Interact

(Source: IMS Content Best Practice Packaging Guide Version 1.1.3 Final Specification)

In simple terms, the “package” shown in the diagram contains all of the physical resources that make up the learning content, and these are listed in the “Manifest”. How this Manifest works and what it looks like are covered in detail in a later chapter. The importance of the Manifest is that it establishes the methods and rules for importing the content into the “Data Store”, and allows the “Run-time Environment” to find, access and display the content as the designer intends. The Manifest is – in this sense – the key to the learning content itself.

As the diagram illustrates, a typical package consists of the Manifest – defined as the Content Packaging Specification – which in turn refers to Metadata, which contains the description of the learning content (eg developer or publisher, topics and so forth), the content’s Organization or structure, and lists all of the physical resources contained in the package. The process of bringing the content together and creating its Metadata is defined as “publishing”. When referring to the *entire process* of bringing together all of the elements required to deliver the complete learning content, including its Metadata, we call this ‘content packaging’.

Before looking in more detail at publishing and packaging learning content, it would useful to briefly define what we mean by learning platforms.

4.1 Learning Platforms – Some definitions

Learning Platform is a generic term used to describe a system of information and communication technologies which delivers and supports learning. So, the term Learning Platform can be applied to, for instance, a Content Management System, a Managed Learning Environment or a Virtual Learning Environment. These are all classes of Learning Platform.

Specifically, a Virtual Learning Environment, or VLE, is defined by Becta and JISC[2] as a software platform which enables learning content to be delivered to learners at many different sites. In addition to a delivery capability, a VLE has five other core functions:

- content is mapped against an appropriate curriculum
- learners can be assessed
- learners' progress can be tracked
- it offers methods of communication (eg a discussion forum)
- it provides tutor support tools.

VLEs – and, in fact, any class of Learning Platform – used for e-learning are designed to support the key aims of SCORM™ (for an introduction to this, see Chapter 7), namely that learning content can be interoperable, reusable, durable and accessible. It is worth noting that the name for a Learning Platform that is very similar to a VLE is Learning Management System, or LMS. This term is typically used outside of the education sector, and is the term used in SCORM™ documents.

Our main focus here is on the VLE and its relationship with learning content. We use the term Learning Platform when discussing information and communications systems in general.

As we have already mentioned, there are many different learning platforms currently deployed throughout the compulsory education sector. Developers of learning content are dependent on these to provide access to their materials, so adopting a standardised approach to content packaging is vital to ensure that content is fully interoperable and accessible across as many learning platforms as possible.

4.2 Learning objects and content packaging

There are several different definitions of a learning object, ranging from anything used for learning to a digital resource that meets curricula requirements and has a particular structure. For the purposes of these guidelines, we have adopted a wide definition of a learning object in terms of its physical content. For instance, a learning object can consist of a single "resource" or it can consist of many "resources" in a package. The proviso is that it only becomes a "learning object" when it has been appropriately published.

Becta recommends that a learning object, on an intellectual level, should consist of three primary elements: a learning objective, content relevant to this objective, and a method of "knowledge check" to enable the learner to assess their own understanding and grasp of the learning objective. It should be a self-contained learning resource. It is also recommended:

- that if the content is packaged, it is packaged and published in accordance with these guidelines
- that it includes run-time communications with the learning platform, such as ECMA script API run-time (see section 7.1).

The means by which the learning platform finds the learning object, and makes it available to the learner in the way that the designer intended, is managed by the Content Packaging Specification, or Manifest.

So, content packaging supports the learning object model by showing how the object's contents fit together, and makes sure that everything required for that content to run is present "in the tin".

As well as learning content, the package can contain Metadata together with other information which serves two main purposes. They offer a description of the object, mapped to a chosen

[2] Virtual Learning Environments – Institutional Readiness

curriculum, so that it can be accurately classified within learning platform search engines, enabling teachers and learners to search for and find it. They also provide the necessary technical “bindings” that enable the learning platform to offer run-time support, for instance, tracking results data and managing bookmarking. To use a simple analogy, let’s consider a tin of beans. In this analogy, the beans are the content; the tin is a standard way of packaging the content. We know how to transport the tin and we know what tool we need to open it. The tin’s label is the Metadata, which tells us what the contents are, who produced it, how much it costs, how much salt and fat it contains, and it has a unique identifier – a barcode.

Because content packaging is both a descriptive and mapping discipline, as well as a technical practice, it is recommended that it is addressed as a key process during the development of the learning object itself, from start to finish, rather than something that is performed once the object has been completed. This timeline presents an overview of the content packaging discipline as part of the content development process.

1	2	3	4	5	6	7
Ongoing Metadata Development Steps 1-5						
Identify/ choose the learning objectives/ curriculum coverage	Develop a Discovery strategy: who is the object aimed at, how will they use it, which other aspects of the curriculum(s) could it support?	Develop the learning approach – structure, media content, how it will be used, to what level will it be disaggregated, IPR issues, etc	Develop/ assemble the object	Finalise and assemble the package description (Metadata) and produce Manifest (Content Packaging Specification) = the “package”	Test in as many learning platform environments as necessary; test your discovery methodology using SCORM test suite	Deliver – re-evaluate over time to ensure that the content packaging remains relevant

This is even more important where there are a considerable number of learning objects to be developed and published, making the packaging and publishing process a disciplined and essential part of the development process from the outset will avoid delays when it comes to delivery. As the diagram shows, it is useful to approach the development of Metadata as an ongoing process.

There may be circumstances where it is simply not feasible or practical to adopt this approach to packaging and publishing, in which case the thought processes and procedures included in these guidelines should be formally addressed as a specific task appended to the end of the development process, and *not* something that is effected as a “quick fix” by one member of the development team. In other words, content packaging is a group discipline, requiring the input and expertise of key members of your content development team.

5. Why do we need best practice guidelines?

The use of ICT for teaching and learning is relatively new, and, in some sectors of education, still in its infancy. However, with thousands of learning objects now available, and many different learning platform solutions, there is a clear need to adopt common best practice publishing and packaging guidelines to ensure that the resources are available when and where they are needed. For the teacher or the learner, the technology needs to be as transparent as possible, removing any barriers between the user and the content.

5.1 Benefits to industry

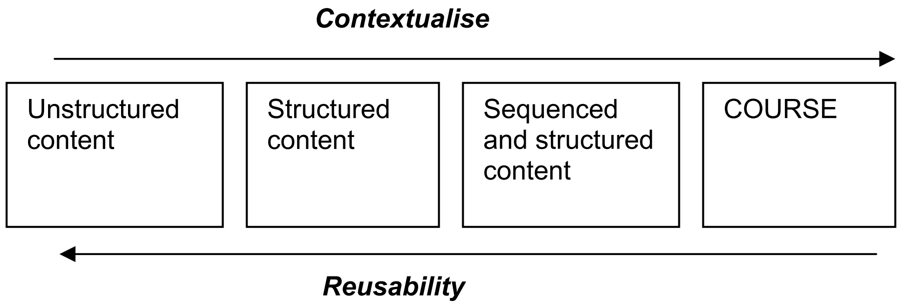
These guidelines do not represent compulsory standards, so their adoption is entirely voluntary. But, with the adoption of a common conformance regime by the learning platform providers, these guidelines, which are consistent with the Becta Learning Platform Conformance Regime, establish the means by which content developers can ensure your content's interoperability across different systems. For some established content providers, this may mean changing the processes that are currently used for content packaging, but the benefits are clear:

- Generates greater interoperability across learning platforms.
- Accurate mapping and classification supports making learning objects easier to find, and more appropriate to the search criteria.
- Reduces the need for individual content package versions to be produced for different learning platforms.
- Establishes more accurate and consistent rules for usage of the learning objects.
- Significantly enlarges the commercial market for your content.

All of which, in a commercial sense, has the potential to reduce time and costs.

5.2 Benefits to education

There is also a clear benefit to be gained by teachers and learners – the users of the content. Where content is more accurately and precisely described and classified, there will be greater satisfaction in both searching for and using the resources. The practice of packaging and publishing learning content affords the learners a degree of personalisation in that they can get the right resources they need, when they need them. For teachers, the practice results in a greater ability to plan and manage their resources, provides greater flexibility and opens up the potential for reusability of learning content. The diagram here visualises this very powerful benefit:



6. What is SCORM™?

6.1 Overview

The Shareable Content Object Reference Model, generally referred to as SCORM™, is a detailed technical framework which has the aim of ensuring that content is:

- Interoperable: content can be shared across different systems.
- Accessible: content is ascribed accurate descriptions so that it can be found.
- Reusable: content can be reused in many different contexts, and its constituents (if allowed by the developer) can be disaggregated for use in different ways.
- Durable: content can be easily updated.

And all of this irrespectively of the learning platform used! It has been developed over a number of years by an organisation called Advanced Distributed Learning (ADL), whose website [http://www.adlnet.org] offers a useful library of relevant documentation as well as news on current issues and debates. SCORM™, as a framework, references internationally accepted standards and specifications developed by IMS and the IEEE, and it sets out detailed criteria for learning platforms as well as for learning objects (SCORM™ defines a Shareable Content Object or SCO) and content packaging. As we will see later, Question and Test Interoperability (QTI) is *mentioned* by SCORM™, but effectively sits outside of its reference.

In 2004, SCORM™ 2004 (version 1.3) was released, and this offers considerable new features over the previous version. The primary differences are covered in the next chapter. SCORM™ is stable but does occasionally get updated to reflect developments in external specifications and changes that are advocated through experience.

The complete SCORM™ framework consists of four key manuals which, since the release of the latest version, are now maintained independently of each other:[3]

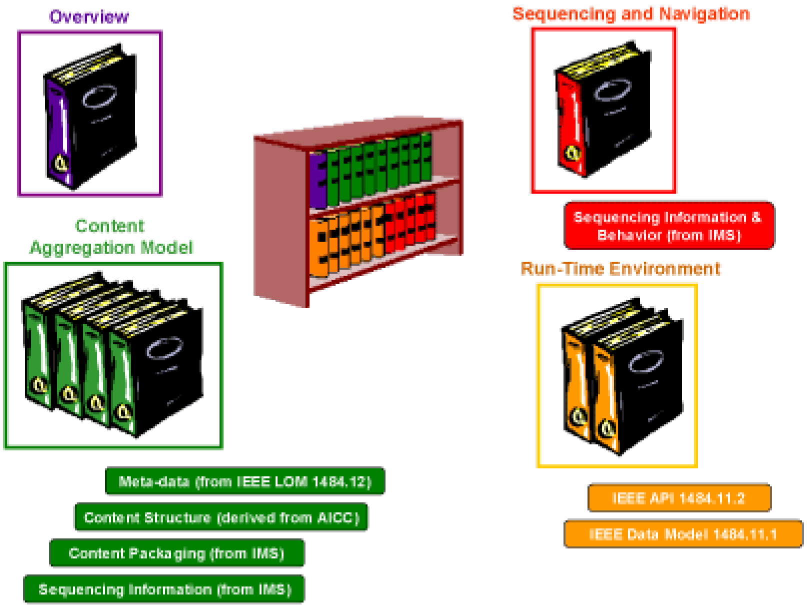
- SCORM Overview
- SCORM Content Aggregation Model (CAM)
- SCORM Run-Time Environment (RTE)
- SCORM Sequencing and Navigation.

A list of specifications and standards that are incorporated in SCORM™ 2004 can be found in Appendix 6.

It is called a “Reference” model because this single model refers to several different specifications. However, it should be noted that SCORM™ also references specifications that were still in draft form, but which are *likely* to be formalised. An example of this can be found in section 7.1 of this document, which focuses on the ECMA Script API.

The following diagram, reproduced from the ADL website, illustrates the relationship between the different elements of SCORM™, and their relationship with IMS and the IEEE.

[3] Shareable Content Object Reference Model 2004 – www.adlnet.org


Overview
Sequencing and Navigation
Sequencing Information & Behavior (from IMS)
Content Aggregation Model
Run-Time Environment
Meta-data (from IEEE LOM 1484.12)
Content Structure (derived from AICC)
Content Packaging (from IMS)
Sequencing Information (from IMS)
IEEE API 1484.11.2
IEEE Data Model 1484.11.1

SCORM™ 2004 has been adopted as a key foundation for the Becta Learning Platform Conformance Regime – with some exceptions. It has also been adopted as the framework for best practice in the development of learning objects and content packaging.

6.2 SCORM™ versions 1.2 and 1.3: Differences

The major difference between this latest version of SCORM™ and its predecessor is something called "Sequencing", often referred to by practitioners as "Simple Sequencing".

Until now, the learning platform has had the responsibility of enabling navigation from one learning object to another, in a non-hierarchical way. So, it is the learning platform that controls external navigation to the object, whereas any navigation within the object itself would be hard-coded internally to the object. In SCORM™, this internal navigation inside the object is called "intra-SCO branching". Internal object navigation should not be confused with "Forward" and "Back" which are normally controlled by the learning platform interface or the browser itself. The SCORM™ view of intra-SCO branching is that it flies in the face of the endeavour to generate content that is reusable and durable, and would certainly limit any desire to enable Disaggregation of the content.

SCORM™ 2004 introduces a framework for the use of Sequencing to establish the rules for navigation from one object to another. So, the addition of Sequencing allows the learning platform to track a learner's progress not just through a single object but a whole sequence of objects. These are known as "behaviours", and they are stored in the Content Package Manifest along with a Content Structure Diagram or Organization. You can, for instance, use behaviours to mandate that a learner must start with one particular object, then do a test and, depending on the results of the test, they can be given access to one or other of the rest of the objects. Be careful not to confuse this new navigation capability with direct navigation between learning objects: SCORM™ does not allow *direct* navigation from one object to another. The learner must be passed back to the learning platform on completion (or whatever rules are established) of one object before navigating to the next in a pre-structured series.

This significant and important step forward allows the learning platform to track a learner's progress through a sequence of learning objects, following a pathway that has been prescribed and defined by the learning designer. Additionally, the sequence of navigation from object to object can vary between different courses. Therefore, the objects are truly reusable, and the learning designer is able to develop far more sophisticated pedagogic solutions for content.

Most of the leading learning platform providers have plans to migrate their services to SCORM™ 2004. Content developers who have already published content should check with their learning platform partners to determine what actions, if any, need to be taken to ensure that the content remains sustainable within these environments. In most cases, no actions will be needed.

More information about Simple Sequencing in terms of content packaging is included in the next chapter of these guidelines.

7. Learning objects: How they should be packaged and published

7.1 ECMA Script Application Program Interface (API)

Some learning objects can communicate with a learning platform, for example, to save results or to indicate that an object was started but not completed. These communications take the form of instructions in a programming language such as Java.

The SCORM™ ECMA Script API is a standardised method for a learning object to communicate with a learning platform such as a VLE when a learner is interacting with the object. The API contains commands, sometimes referred to as SCORM calls, which instruct the learning platform on how to manage the object, for instance, the call GetValue can be used to call up the learner's name and present it on the screen. The API is, in this sense, the key to the learning platform's Run-Time Environment, and is the method by which the content communicates with the learning platform, enabling the learner's actions to be tracked.

In the learning object, the API calls are included in the object in the form of JavaScript code and it is these which enable the object to be called and delivered to the browser via the learning platform, and for the object and platform to communicate with each other.

Becta's guidelines recommend the use of SCORM™ API for communications and interactions between content and learning platforms.

As the commands contained in the API are intrinsically bound with the pedagogical approach designed into the learning object, they are an important feature of the package. Hence, simply instructing the learning platform to run the object, or record that the learner has completed it, may not satisfy the full requirements of pedagogical best practice.

The Run-Time Environment (ie the learning platform) data model that SCORM™ uses, cmi.core, takes as its basis the Data Model for Content Object Communication, referenced as IEEE 1484.11.1.(At the time of publication, this has just become an accredited standard.)
A list of API calls and elements that learning objects should support is included in Appendix 4.

Learning objects that do not use these API calls provide limited ability for the learning platform to control the learner's experience of its contents other than launching the object. For instance, the learning platform will consider that the learner has completed the learning object, even if the first screen is the only one actually visited. Also, an object that does not use any of these calls is not considered to be a SCO in SCORM™.

More detailed information can be found in Appendix 4.

7.2 Manifests and sub-Manifests

Each content package must include a top-level Manifest file (or IMS Manifest File), which should always be named "imsmanifest.xml", and always be presented in lower-case characters. To clarify, this file must be placed at the root of the Package Interchange File (eg .zip). Without the presence of the imsmanifest.xml, and in this location, the package and its contents cannot be unpackaged and used within a learning platform environment.

This special XML file describes the package itself through the following information:

- Metadata
- Organization
- Resources.

Metadata and Organization are dealt with in detail in subsequent chapters. The Resources section of the Manifest is an XML element containing references to all of the actual resources and media elements that make up the learning object. The Resources section can also include the Metadata describing the resources and any references to files held externally. Note that, whereas in IMS version 1.1.3, there is a reference to Resources which are not included in the package but which are referenced using URLs, this is not a recommended practice in SCORM™ 2004. From a practical perspective, resources held at different URLs may not be under the control of the content developers, and so their availability may be at risk.

The simplest structure, one which is adopted by most providers, is to have a single Manifest in a content package. However, it may be useful in some circumstances to have sub-Manifests. It is recommended that, where there is a need to contain sub-Manifests, that these are held "side by side" within an otherwise empty learning object package. When planning a collection of resources, you should consider the structure that is most appropriate for your needs. For example, where a content package contains a large number of learning objects, such as a course, then sub-Manifests could be used for each module of the course.

Content Packaging Tools will automatically generate the imsmanifest.xml as the top-level Manifest in the package, and will allow you to customise it by, for instance, adding in your content files. A discussion of Content Packaging Tools can be found in the next chapter.

In these scenarios, we have assumed that the contents of the learning object are included in the package, but there is another model that it is worthwhile mentioning here. You can have packages where the Metadata links to content which is held and controlled elsewhere, at another URL, for instance. So, it's not always the case that the learning content needs to be packaged up with the Manifest and Metadata.

The Manifest file is, then, precisely that: a complete manifest of everything concerning the learning content and how it should be accessed and used. It must list all of the files that make up the object's content, whether this is held in the package or elsewhere. Even the smallest omission could result in your content not running as intended.

7.3 Organization

The Organization, or Content Structure Diagram, is one of the three key elements contained in a content package's Manifest. It is a description of how the contents of the learning object are to be presented to the learner, including their presentation order and any particular Behaviours. This clearly has a direct bearing on Sequencing and Behaviours.

Each Manifest must contain an Organization. So, where a learning object is very simple, perhaps only containing a single resource such as a video clip or a picture in a web page, its Content Package Manifest must still contain a reference to Organization: <organizations/>. This is known as an empty element.

Essentially what your Organization should do is organise the learning object's contents into a tree structure that complies with the pedagogical approach taken by its designer. In this way, the Organization is used to present and control the hierarchy of the contents. The second element contained in the Organization is a list of the Behaviours that are ascribed to the object's contents: Sequencing and Behaviours, which are covered in more detail in the following sections. Suffice to say that Behaviours describe what should happen (in non-technical language) and each is related to a SCORM™ function, which is the code that enables the Behaviour to occur.

In the light of this, it is clear that before an Organization can be developed as part of the package's Manifest, the strategy for Sequencing (if any is required) must be fully developed. Ascribing Behaviours and tree structures to a learning object's contents *after* it has been developed as a complete object, could result in a confused experience on the part of the learner, and a confused pedagogy.

An example of an Organization is:

```
<organizations default="TOC1">
  <organization identifier="TOC1">
    <title>Key Skills IT Level 3</title>
     <item identifier = "ITEM33" identifierref = "RESOURCE177">
       <title>Quick test in Information Technology</title>
    </item>
  </organization>
</organizations>
```

The other primary aspect of the Organization is its Aggregation, which is the topic of the next section.

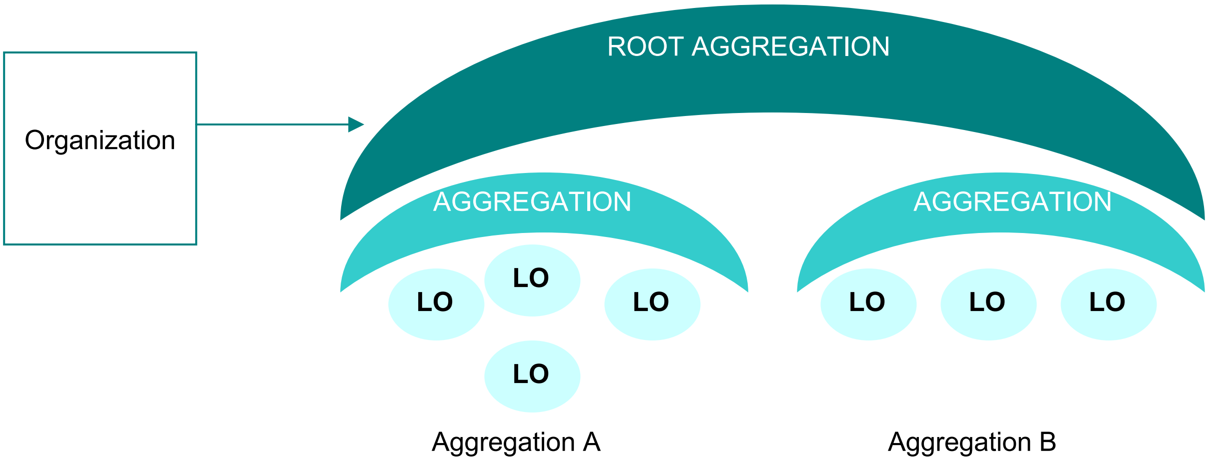
7.4 Aggregation

Aggregation – meaning the sum total of multiple disparate elements – is actually best approached from the bottom up. By this we mean, approached from the perspective of Disaggregation. When designing a learning object, of whatever size, it is important to consider to what extent you will allow a learner, or a teacher, to "disaggregate" the content for use in other contexts and with other resources.

This ability to extract an object's "granular" content is highly desirable for the content users, but it does have commercial implications for the developer, especially in terms of Intellectual Property Rights (IPR). So, from the outset of development, this issue must be addressed. There is a category in the Metadata – "rights", which is specifically reserved for storing information regarding IPR, terms of use and any restrictions. It is important that this is accurately and meaningfully addressed for the protection of both the content owner and the content user.

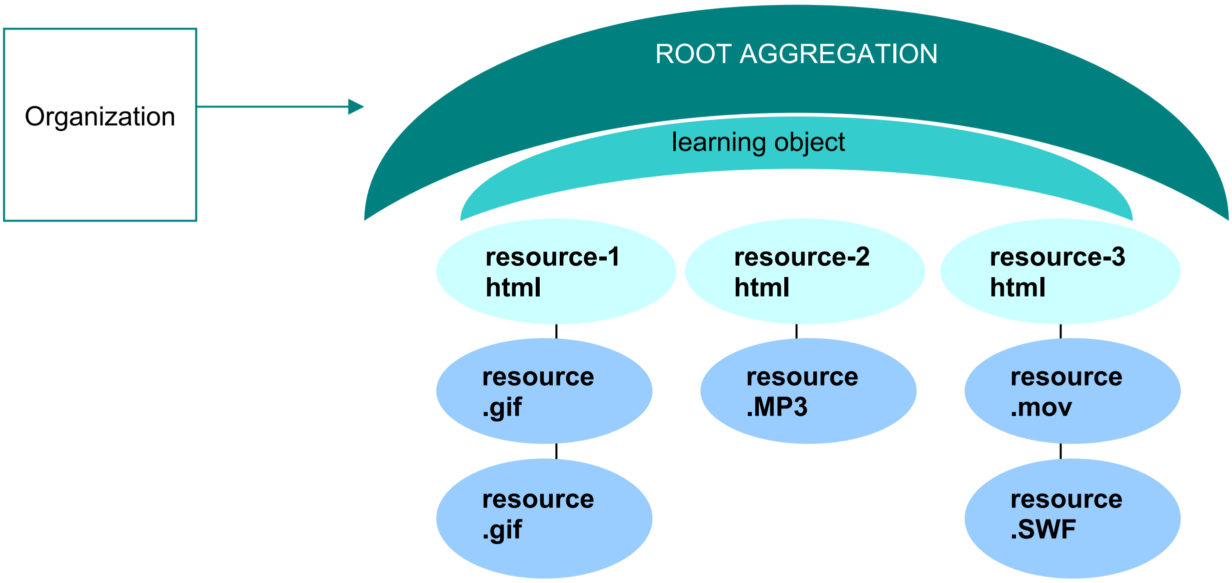
The use of the term Aggregation assumes that there are multiple entities involved. For the purposes of clarity, a content package can contain many learning objects – an Aggregation, and a learning object can contain many resources at granular level, which can also be referred to as an Aggregation.

In SCORM™, an Aggregation is defined as a parent and its children held in a hierarchical tree structure. Aggregations are used to group related content together. The diagram below shows a simple example. The Root Aggregation itself is empty but beneath it there are two Aggregations.

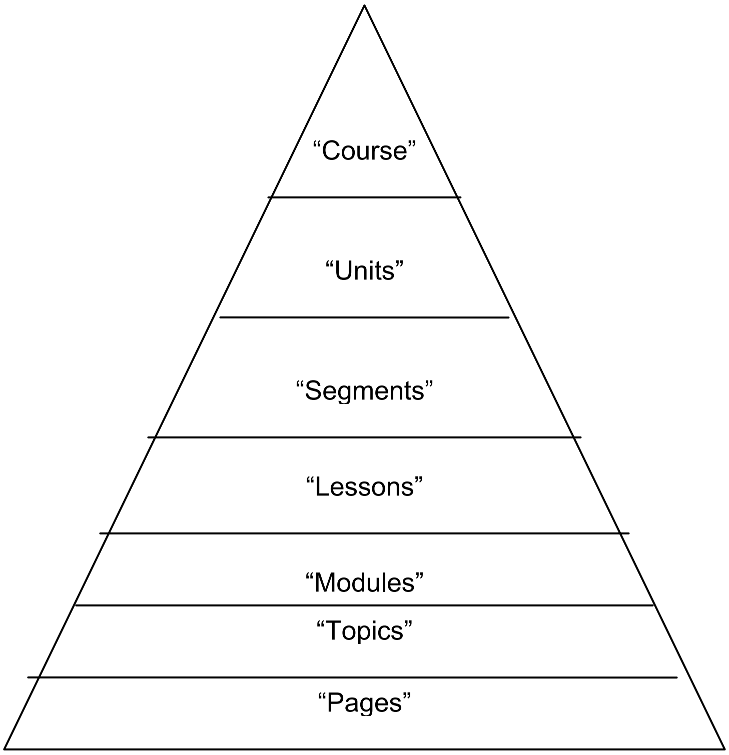


In this example, the content package contains two distinct but related modules of learning objects. Because the objects are naturally best grouped into modules, each has its own Aggregation, which sits under the Root Aggregation. The Organization would define the order in which the Aggregations and individual objects are presented, and any associated Behaviours.

The Organization should always start at the Root Aggregation, and flow down to the lowest level that the developer wishes to allow the content to disaggregate at. The next example shows a straightforward learning object as a single entity containing a number of assets, with the intention of allowing complete Disaggregation.



As we have previously mentioned, a learning object, Aggregation or Root Aggregation can represent any number of components, so when designing the Organization or Content Structure Diagram, particularly for a complex package consisting of many components, it is useful to determine a framework of meaningful terms to denominate each incremental component. For example:



SCORM Best Practice Guide for Content Developers[4] includes some good and useful templates for Aggregations within Organizations.

“The content developers are responsible for deciding whether or not to use (sub)manifests when creating content packages. One rule of thumb is to use a single manifest for tightly coupled content where no part of the content organization may be presented out of the context of the aggregation. Content developers may want to create separate manifests ((sub)manifests) for each lesson, module, etc. This is entirely up to the content developer.”

(Source: Sharable Content Object Reference Model (SCORM) Content Aggregation Model Version 1.3.1, Advanced Distributed Learning (ADL), 2004)

When planning and organising the Organization and Aggregation(s) of a content package, you must always bear in mind that the Manifest is effectively a device for pre-loading data via the learning platform.

7.5 Metadata

Metadata is as vital an element in the content package Manifest as the Organization. It works in a similar way to a library index card system, or filing system, and contains details on the object’s subject, contents, author and copyright status as well as other key indicator information.

It is included in the form of XML. The SCORM™ Metadata Schema references the IEEE Learning Object Metadata (LOM) version 1.0. Becta’s best practice guidelines advocate the application profiles of this standard, with extensions concerning curricular classifications.

[4] © 2003 Carnegie Mellon University

To start with, it is recommended that you assign a unique identifier to your content package. For example, Curriculum Online uses three types of identifier: DOI, ISBN and URLs. Other examples include URN, GUID and HANDLE. Whichever type you use, it is recommended that identifiers are "resolvable", that is, you can enter them into a browser and they will connect to an intended location.

The package's identifier is stored in the Manifest, but, importantly, each Metadata record should also have its own identifier. The identifier provides a unique identification for your object within an Aggregation, and is also essential for cataloging and commercial purposes. Including an identifier ensures that a single object can be aggregated with more objects without any problem, and it allows the objects to be managed properly, since managers know exactly which objects they have got!

In the SCORM™ model, there are nine Metadata categories (LOM 1 to 9), each of which has subcategories. Various implementations of LOM recommend the minimum categories that have to be completed and leave it to the content provider to give as much or as little extra detail as necessary. Remember: the more Metadata categories that you choose to include and the extent to which their subcategories are completed, will influence the way content is found and displayed within learning platforms.

Another practical guideline is that, before developing the Metadata, it is important to understand the search habits or trends and vocabulary that are common to those who your learning object is aimed at. We look in more detail at this in a later part of this chapter. Also, consider carefully the style of terminology that you use in your Metadata: technical descriptions such as "virtual environment" or "interactive simulation", for example, may hold no meaning for those who might be searching for your object. Avoid too much emphasis on "marketing" or "sales" orientated language, even if your learning object is for sale. This kind of language can cloud the real purpose of your object.

The nine basic Learning Object Metadata (LOM) categories are:

General
A general description of the package including Product Code, Title, Language, Description, Keywords or Phrases, Coverage, Structure and the level to which the content is aggregated such as Lesson or Topic.

Life cycle
Publishing details including publisher's name, author's details (if different), version information and lists of contributors.

Meta-Metadata
Includes a ProductID, which is a unique identifier for the Metadata record itself. In the Curriculum Online version, this consists of SupplierID-RecordID, with the SupplierID provided by Curriculum Online. This is also used to store information on the Metadata Schema that has been used, for instance, Curriculum Online Schema version 1.1.

Technical
This is used to store technical information on each asset in a learning object. Its format is identified (eg .GIF), its size, its location, resource requirements, installation remarks, requirements for other platforms, and the asset's duration.

Educational
This element allows you to describe the educational values and approaches (active, expositive, mixed or undefined), the type of learning resource (eg assessment, open activity, etc), methods of delivery, levels of interactivity, who the resource is aimed at (eg teacher, learner, etc), educational level (eg primary, secondary, etc), the typical age range the resource is intended for, difficulty levels, typical learning time, and a description of how you intend the resource to be used. This last element compliments the resource description placed in the General section, and can be used, for instance,

to state that the resource is designed to be used by teachers with a whiteboard in a group activity, supported with handouts for learners.

Rights
This is where the content developer can store information on the cost (or otherwise) of the resource, and the all-important information regarding copyright. It also contains an element for describing any available support, for instance, technical support and support for teachers.

Relation
This is an option to describe any relationship with other available learning resources. For example, your learning object may contain a reference and a URL link to data stored outside of the learning resource.

Annotation
This is an option where you can offer information provided by an independent source on your learning object, such as a third party review or teachers' comments.

Classification
This is where you classify the content of your resource against a curricular framework (that is, according to the topics that it covers, and any teaching or learning frameworks that it is suited for). This is an important element in Metadata tagging, and is covered in more detail in the next section.

7.5.1 Classification (Taxonomies)

Classification systems, or taxonomies, are the means by which content developers can classify learning objects to specific elements of an educational curriculum.

There are different curricula for different educational systems globally. However, and this is important, if learning objects are intended for use in education in England, *you must apply the Curriculum Online classification***.** The Curriculum Online taxonomy is used for defining products for the Curriculum Online website. The Metadata used in content packages and for distribution of learning resources should be based on this taxonomy, but you may use different parts of the taxonomy for various levels of your content's levels of granularity.

The quickest way to familiarise yourself with this particular taxonomy is to download the Curriculum Online tagging tool from the Curriculum Online website (see Appendix 3). As a quick overview, this is how it works:

- Select a subject from the list presented, eg art and design.
- Select a key stage and a year, eg Key Stage 1, Years 1 and 2.
- Select from A National Curriculum Programme of Study, List of Topics or Schemes of Work.

This selection opens up further detail to enable you to drill down to the lowest level. For instance, if we select Programme of Study, Art and Design, Key Stage 1, we are able to drill down through:

Exploring and developing ideas…
  A record from first-hand observation…
    Collecting visual and other information
    Exploring and developing ideas
    Ideas/values/beliefs/meanings
    Generating ideas

It is important that you select subjects that the learning resource is specifically intended for, avoiding tenuous associations, although you can specify a resource as having the potential for teacher-specific, cross-curricular skills.

Although it is possible to “tag” or associate your learning object to more than one element contained in the Curriculum Online Metadata application profile (an application profile is a localised version of a schema such as LOM), it is good practice to limit this to a maximum of four.

As we said earlier, the Curriculum Online application profile is an evolving specification and so it is good practice to regularly review the classifications made for any existing learning objects to ensure that their associations remain relevant and appropriate. Incorrect or out-of-date Schema decorations and locations (ie their associations) are one of the most frequently observed sources of difficulty in locating the right learning resources when they are needed. As a final check, it is also good practice to consult with a professional educationalist who is experienced and knowledgeable in curriculum structures. Informed advice from a librarian can also be extremely useful.

The precision and accuracy with which you tag your learning objects to Metadata Schema, whether Curriculum Online or any other, will directly affect the ability of the learners and teachers to locate it and use it.

7.5.2 Descriptions

People looking for learning content generally search by using one of two methods: they will either search by curriculum-based topic or subject, or they will use key words in a free text search. This latter has a bearing on what words are placed in the description and keyword fields in the General element of the Metadata.

Consider the following extracts taken from the descriptions of two different history resources:

- A. Using photographs and extracts from factory inspectors’ reports, this resource considers the impact of the 1833 Factory Act. Background information and teacher’s notes are included.

- B. It is not always easy for schools to provide authentic visits. This multimedia history resource gives the feeling of visiting a special site without major expense.

Extract A references a specific event – the 1833 Factory Act. It also offers clear information on what else is included, and more particularly, what is available to support the teacher’s use of this resource, and it succinctly describes the approach taken in the learning object.

Extract B provides no context-specific information at all.

Descriptions should be limited to around 30 words, and should feature the facts. Is the resource concerned with a specific event, person or place? If so, ensure they are named. Using simple terms, explain the approach taken – photos, maps, audio, and so on – so that the teacher or learner has a clear picture of what the resource will include. For instance, the use of particular media such as audio or high impact visuals may affect a teacher’s choice of resource. Remember that all of the other information associated with the learning object, such as curriculum classification and technical details, are stored elsewhere in the Metadata record.

In the Keyword element, concentrate on precisely that – the Keywords. Also, where a word has two or more widely accepted spellings, use them all.

Note that people may also search by the name of the resource supplier. Ensure that all details concerning the learning object developer/author/publisher are correct and accurate.

7.5.3 Intellectual Property Rights

Intellectual Property Rights (IPR) matter for a number of reasons. First, the content developer needs to be assured that their interests and property are protected, particularly where the learning objects are made commercially available. But, the content developer must also ensure that the rights to any content used in the learning object which has been provided or obtained from a third party are properly dealt with. Equally, the content developer must clearly specify to what extent your learning objects may be disaggregated, allowing the assets to be used in different contexts, or used to make up new learning objects by, for instance, a teacher.

- Ensure that the IPR in any content that is not your own is administered and audited.
- Ensure that you use the Metadata Rights element to state clearly the usage that you will allow.
- Ensure that you have dealt with the Aggregation/Disaggregation issue clearly and accurately.

7.6 Compliance and Conformance

The terms “Compliance” and “Conformance” are often confused to mean one and the same thing. The distinction between their respective meanings does, however, become important when they are used in reference to standards and specifications. “Conformance” implies a strict adherence to a set of formal standards or specifications, within a certification scheme. “Compliance” means a *tendency or intention* on the part of the content developer to follow established specifications or standards.

Although there is the Becta Learning Platform Conformance Regime, there is no similar regime that is applicable to content. You can, however, determine your content’s compliance with SCORM™ by downloading the SCORM™ Test Suite, available freely from the ADL website (see Appendix 3).

Testing learning objects in the environment for which they are intended is a critical process in the development life cycle, and sufficient time should be set aside for this task. It is strongly recommended, where your content is destined to be hosted by specific learning platforms, that you organise comprehensive testing in these environments prior to formal delivery. Also, it is worth noting that CETIS runs regular “code bash”, events where content developers can try out their content on a variety of different learning platforms.

7.7 Delivery

Becta recommends that all learning objects are packaged using a Package Interchange File (PIF). This should include the top-level Manifest and all associated resource files, which are identified by the Manifest. In this format, the package can be transported in any way that the content developer chooses.

The recommended way to construct a package is to zip up the contents using a PIF (eg PKZip v2.04g [.zip]). Some learning platform partners will specify a maximum file size for content packages, typically 100Mb.

It is also useful to have some housekeeping rules for file names, paths and formats. Typically, it is recommended that file names have the following characteristics:

- always in lower case and alphanumeric
- set a maximum length (the longer the file name, the more margin for error!)
- use underscores instead of spaces
- no two physical files should have the same file names, or the same pathways.

7.8 Discovery

Earlier in these guidelines, we stressed the importance of understanding the search habits, trends and vocabulary of the teachers and learners who would be searching for learning objects. Clearly, taking advice from an expert in your targeted curricula is essential.

The starting point is to think of all the possible audiences who might be interested in your learning content: who are they, how might they use the content, in what context, is the content cross-curricular, what words might they use when carrying out a search? In other words, it is good practice to develop a Discovery strategy, and this is something that should be initiated at the start of the content development. Discovery strategies can be structured and categorised around the curriculum (remembering that the curriculum Metadata Schema may not be an exact replica of the actual curriculum itself), as well as key words.

7.9 Future: Sequencing and Behaviours

In Chapter 6, we discussed Sequencing and Behaviours as the major new ingredient to SCORM™ 2004. But what does this really mean? And what will it mean for content developers and learners? First, a couple of definitions.

SCORM™ refers to two methods of navigation within a content package. Intra-SCO*,* or learning object Branching**,** refers to the navigation that is embedded or hard-coded into the learning object itself, and which is disassociated from the learning platform. In other words, the learning platform has no control over what goes on *inside* the object. Other key rules are:

- in a SCORM™ environment, a learning object cannot link directly to another learning object
- navigation between learning objects is managed by the learning platform
- a learning platform will only launch one learning object at a time.

Inter-SCO, or learning object Sequencing, is the means by which the learning platform determines the order – and event criteria – for navigating between learning objects. The most common sequence events are:

- rules for what actions the learner must have taken to complete the Root Aggregation (ie the total contents of the Content Package)
- rules for what actions the learner must have taken in order to access any particular learning object in an Aggregation (eg passed a test)
- rules for what happens if a learner fails a test (eg a learning object that they have already completed is re-presented)
- rules for what happens if a learner passes a test (eg they might not have to engage with a particular learning object).

These basic rules are often referred to as Simple Sequencing.

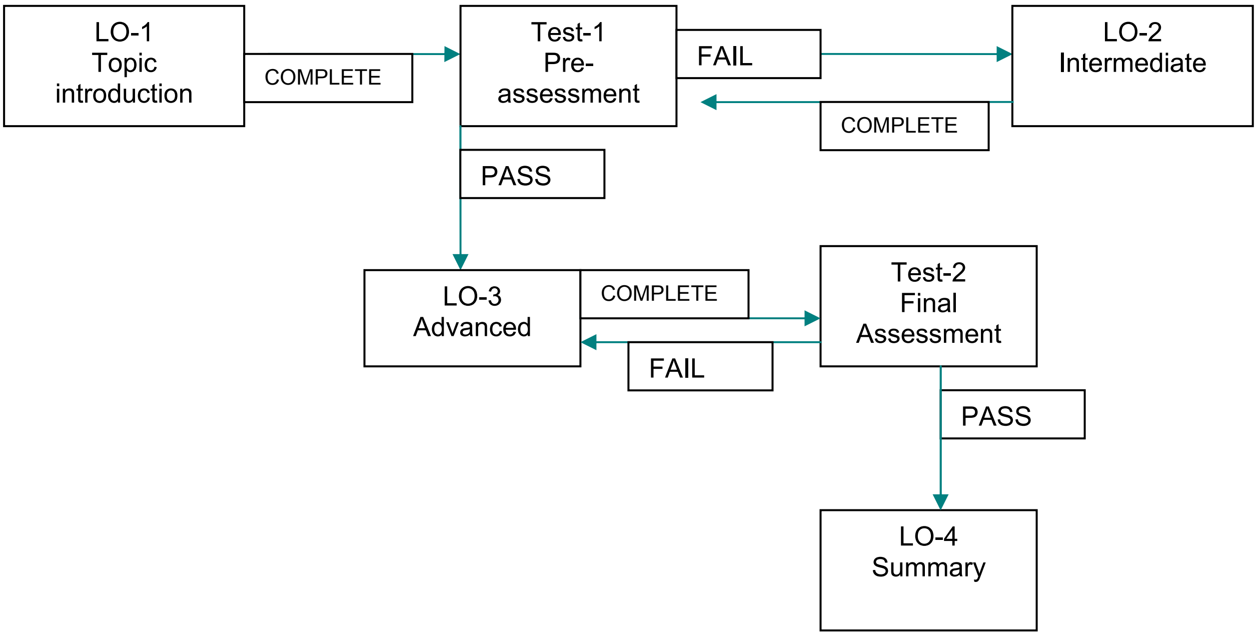
So, Sequencing can be used to establish the pass/fail criteria, to determine a pathway through a number of learning objects, and to use pre- and post-assessment or testing to determine whether or not someone needs to engage with the learning content, and, finally, to determine the pathways for any remedial learning that may be needed by an individual learner.

Exactly the same effect can be achieved inside the learning objective itself – using intra-SCO branching – but the difference is that in this case, the learning platform is ignorant of the actions and performance of the learner. What Sequencing does is enable the actions of the learner to be guided, mediated, and their performance in tests not only captured and stored, but also used to dictate individual learning pathways. In so doing, the use of Sequencing and

Behaviours introduces the ability to *personalise* the learning experience for individual learners, and place the learner firmly *at the centre of the learning experience.* These are two very key themes in modern compulsory education.

The impact of Sequencing on the learning experience means that the terminology and logic used in creating the Behaviours and SCORM™ functions that make up the Sequencing rules must be correct.

For the content developer, the use of Sequencing may require the development of new skills and practices. In particular, the designer of the learning objects needs to become very familiar with the Behaviours that are available. However, one way of looking at Sequencing is to see it as the ability to do outside the learning object what you can already do inside – albeit with some limitations. But the advantages are clear. Let's take a simple example:



In this structure, with the arrows representing interactions with the learning platform, the learners start at LO-1 and, when completed (which can be defined in the Behaviours), the learner is passed to Test-1, to check their knowledge and understanding. If this is failed, the learner is passed onto a new object, LO-2, for remedial or intermediate learning, whereas the learner who has passed navigates to LO-3, offering a more advanced level. On completion of LO-2, the learner is passed back to Test-1 to ensure that they now have sufficient knowledge and understanding to progress onwards. On completion of LO-3, learners are passed to Test-2 to check their knowledge. A pass at this point navigates the learner to a final Summary, LO-4, whereas a fail would take the learner back to LO-3 to refresh their knowledge and understanding. This can, of course, be made more complex and sophisticated with discrete and special objects purely for remedial purposes, and so forth.

Now this could have been undertaken inside a single learning object, but, in this model, the learning platform is controlling the navigation, and therefore the learning experience, and this experience is tailored to the needs and abilities of the learner.

8. Content packaging and tagging tools

There are many readily available – often free – tools designed to generate the required Manifest and Metadata XML to form content packaging for a learning object. In fact, many of the learning platform providers have their own customised tools. This might seem to be a simple solution for the content developer, but before opting for this method, there are certain issues that you ought to consider.

First, and most importantly, ensure that any tool you propose to use will generate content packaging – particularly the Manifest – that is compliant with IMS 1.1.3[5] which, as we mentioned earlier, is referred to in SCORM™ 2004. It is also good practice to use the same tool for all of your content in order to ensure that the packaging is consistent across, for example, a series of objects. A list of URLs where the most commonly used and freely available content packaging and tagging (publishing) tools can be accessed is contained in Appendix 3.

Not all of the tools will generate the entire complement of information needed to complete the content packaging. For instance, a tool may offer an automated process of generating the Manifest including the Organization and Aggregation/Disaggregation rules, but may not offer a solution for producing the Metadata tagging essential to map your content to the required curriculum taxonomy.

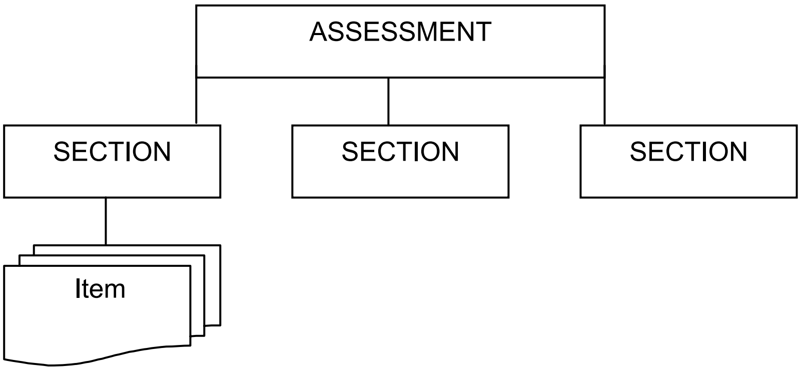
It is a matter of preference. A content packaging or tagging tool can speed up the packaging process, and limit the margin for error in coding. However, the same principles for accuracy and precision apply: you will still need to give careful thought to how your content will be used, by whom, in what context, key words for free text searching, and the level to which you allow your content to be disaggregated in a learning platform, or not. And, for Metadata tagging, you will still need specialist curricula knowledge to ensure that your content is tagged correctly and appropriately to your selected taxonomy or taxonomies.

[5] IMS content packaging data and model binding specifications, version 1.1.3

9. Question and Test Interoperability

Content packaging protocols and procedures for tests work in exactly the same way as they do for learning content, and they are packaged as learning objects. The key structures are:

- Assessment, which is the basic test unit.
- Section, which is a container for groups of sections and items that support a common learning objective.
- Item, which is a self-contained question/response element in which individual questions are contained.



SCORM™ does not provide any framework for *how* question and test objects are designed, scored or structured. What it does do is support the use of "score", "passed/failed", "mastery status" and "time spent" for each individual object. This is distinct from Question and Test Interoperability (QTI), which is the most commonly adopted standard. The important rule to bear in mind is that the learning platform will only track a learner's activities at the level of the learning object. So where, for instance, you want to create a bank of 10 questions, and know the learner's performance for each, you would have to create each question as a separate learning object.

10. Accessibility

Standards for accessibility are evolving all the time and, ultimately, with such diverse uses and users of the web, conflicts between standards will be inevitable. Content developers are expected as far as possible to comply with current approved guidelines and standards to enable accessibility to learning content. Potential learners will include those with physical, language or cognitive disabilities, and those who are visually impaired or blind, as well as people who are hard of hearing or deaf. Some learners have multiple disabilities.

Content developers are advised to consult the Becta website for current information on accessibility guidelines and standards:

http://www.becta.org.uk

IMS has developed an accessibility profile, consisting of two parts. Accessibility for the Learner Profile provides the means for storing learners' accessibility preferences. This specification is mirrored in IMS Accessibility for Metadata, which allows developers to describe the accessibility of their content. Using both specifications, learners can be matched to the most appropriate content for them. More information on this can be found at the following URLs:

http://www.cetis.ac.uk/members/accessibility/index

http://www.techdisc.ac.uk

Appendix 1

Glossary

ADL	Advanced Distributed Learning, developers of SCORM™
Aggregation	An Aggregation is defined as being a ‘parent’ and its ‘children’ in a tree structure, and is held in the Organization element of the content package’s Manifest
API	The Application Program Interface is a standardised method for a learning object to communicate with a Learning Platform
API calls	Application Program Interface calls – agreed set of commands used to effect communications between learning objects and learning platforms
Becta	British Educational Communications and Technology Agency
Behaviour	The actions which a piece of software are expected to do, given certain commands, or in a certain context
CETIS	Centre for Educational Technology Interoperability Standards
GUID	Globally Unique Identifier, which can be a DOI, ISBN or URL
IEEE	Institute of Electrical and Electronics Engineers
IMS	Instructional Management Systems
IPR	Intellectual Property Rights
JISC	Joint Information Systems Committee
LMS	Learning Management System, similar to a Virtual Learning Environment, and which is a class of learning platform.
LO	Learning Object, meaning a structured collection of learning resources that collectively have an educational value, and which is stand-alone
LP	Learning Platform, meaning a system of information and communications technologies used to deliver and support learning
Manifest	A Manifest is a description of everything that is contained in the content package, including its Organization and Resources
Metadata	Metadata is used to describe learning content, including information on its developer, version, and relevance to curricula elements
Organization	The Organization describes the entire structure of the learning content, whether this consists of one or more individual learning objects
PIF	Package Interchange File, eg ZIP
SCA	Shareable Content Asset (SCORM™ term), meaning the individual media elements such as web pages, photos, video clips, etc,

	contained in a learning object, or as stand-alone content elements.
SCO	Shareable Content Object – a collection of assets that becomes an independent, defined piece of instructional material. SCOs are the smallest logical unit of instruction you can deliver and track via a learning management system. They may contact single learning objectives, collections of learning objectives, tests, scenarios, simulations, etc[6]
SCORM™	Shareable Content Object Reference Model, a detailed technical framework that references internationally accepted standards and specifications for learning platforms and learning content
VLE	Virtual Learning Environment, a class of learning platform

[6] *SCORM Best Practice Guide for Content Developers*, Carnegie Mellon University, 2003

Appendix 2

Packaging and publishing checklist

This checklist is designed to provide you with a quick and easy reference to help with packaging and publishing. This is by no means an exhaustive list, but it does contain some of the key steps, and can easily be adapted to suit your own circumstances.

Resource	
Decide on the granularity of the learning resources	
Will any dependent resources be used?	
Will your resources use any external links such as third party URLs?	
How will your learning resource be structured and organised?	
Decide on your Aggregation strategy	
Develop a Discovery strategy	
Resource files	
What files are needed for each learning resource?	
Organise the content so that the files for each resource are in a separate folder	
Organise approach to file naming conventions	
Organise files according to the targeted user, eg teacher, learner, etc	
Metadata	
Research and develop resource descriptions in accordance with relevant curricula taxonomies	
Prepare relevant key words of each resource	
Address any copyright issues, and ensure that these are accurately reflected in the Metadata	
Ensure that descriptive information and learning times are checked and accurate	
Packaging	
Select an appropriate name for the package	
Obtain globally unique identifiers	
Make up the Manifest, and ensure that all resources are listed accurately	
Ensure that the Aggregation and Organization data is accurate	
Check file names	
Organise thorough testing of the package in appropriate learning platform environments	

Appendix 3

Content packaging and tagging tools

The following is a list of URLs where the most commonly used and freely available content packaging and tagging tools can be downloaded.

The Curriculum Online tagging tool is a publishing tool, which means that it is focused on building the Metadata elements for the content package:

http://www.curriculumonline.gov.uk/SupplierCentre/taggingtool.htm

The reload tool, developed by CETIS, is, by contrast, predominantly used to build the Content Packaging Specification or Manifest:

http://www.reload.ac.uk

Useful URLs

Becta
http://www.becta.org.uk

Curriculum Online
http://www.curriculumonline.gov.uk

CETIS
http://www.cetis.ac.uk

Advanced Distributed Learning
http://www.adlnet.org

IMS Global Learning Consortium
http://www.imsproject.org/content/packaging

IMS Question & Test Interoperability Specification
http://www.imsproject.org/question

IEEE LOM Metadata
http://www.itsc.org.sg

All website links, here and throughout the document, are accurate at the time of publication.

Appendix 4

Run-Time Environment API calls

The following is a list of API calls and elements that learning platforms must comply with under the terms of the Becta Learning Platform Conformance Regime. Where relevant to the aims of learning objects, objects should also support these:

- conform to the IEEE P1484.11.2-2003 ECMAscript, in particular be interoperable with the SCORM^{TM} 2004 Run-Time Environment API calls:
  - Initialize
  - Terminate
  - GetValue
  - SetValue
  - Commit
  - GetLastError
  - GetErrorString
  - GetDiagnostic
- conform to the IEEE P1484.11.1-2004 data model and support the following elements in Table 1.

Table 1: SCORM Run-Time Environment data model elements summary

Data model element	Description
Comments from learner	Contains text from the learner
Comments from LMS	Contains comments and annotations intended to be made available to the learner
Completion status	Indicates whether the learner has completed the SCO
Completion threshold	A value against which the measure of the progress the learner has made towards completing the SCO can be compared to determine whether the SCO should be considered completed
Credit	Indicates whether the learner will be credited for performance in this SCO
Entry	Contains information that asserts whether the learner has previously accessed the SCO
Exit	Indicates how or why the learner left the SCO
Interactions	Defines information pertaining to an interaction for the purpose of measurement or assessment
Launch data	Provides data specific to an SCO that the SCO can use for initialisation
Learner ID	Identifies the learner on behalf of whom the SCO instance was launched
Learner name	Represents the name of the learner
Learner preference	Specifies learner preferences associated with the learner’s use of the SCO
Location	Represents a location in the SCO
Maximum time allowed	The amount of accumulated time the learner is allowed to use a SCO in the learner attempt
Mode	Identifies the modes in which the SCO may be presented to the learner
Objectives	Specifies learning or performance objectives associated with an SCO
Progress measure	A measure of the progress the learner has made towards completing the SCO
Scaled passing score	The scaled passing score for an SCO
Score	The learner’s score for the SCO

Data model element	Description
Session time	The amount of time that the learner has spent in the current learner session for the SCO
Success status	Indicates whether the learner has mastered the SCO
Suspend data	Provides information that may be created by an SCO as a result of a learner accessing or interacting with the SCO
Time limit action	Indicates what the SCO should do when the maximum time allowed is exceeded
Total time	The sum of all of the learner's learner session times accumulated in the current learner attempt prior to the current learner session

(Source: Sharable Content Object Reference Model (SCORM) Run-Time Environment Version 1.3.1, Advanced Distributed Learning (ADL), 2004)

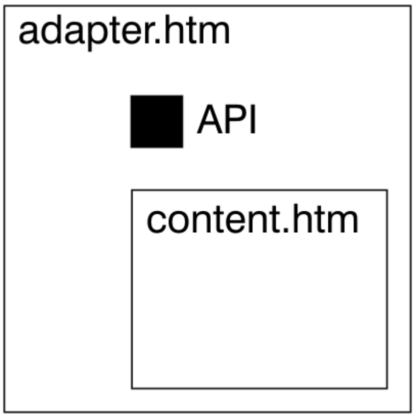
Providers should be able to provide details of how they meet appropriate SCORM self-certification tests. Additional informative guidance can be found in the document *SCORM Run-Time Environment Version 1.3.1*.

Use of the SCORM run-time calls

The following is intended to provide a brief overview to help understanding of some of the key benefits offered by the SCORM run-time calls. It includes code snippets which may be useful, but if you are actually implementing the run-time you should refer to full technical documentation available at www.adlnet.org.

Finding and initialising the API

A learning platform will normally launch a learning object by opening it in a frame within a page called the adapter. This page is provided by the platform. In the following diagram, *adapter.htm* is part of the learning platform, and *content.htm* contains the learning object. Note that the relative size of the two boxes is not significant – it is up to the learning platform to determine how much screen space (if any) is occupied by *adapter.htm*.



As well as the content, the adapter also contains an object called API which can be used by the learning object to communicate with the learning platform. Whenever the content is loaded, it should therefore determine whether it has a parent frame which contains an object called API – this is achieved using code along the following lines:

```
var API;

function onLoad () {
```

```
          API = null;
          if ( (window.parent != null)
            && (window.parent.API != “undefined”) ) {
                API = window.parent.API;
          }
     }
```

If a learning object finds the API, it can get and set information, using the following five key functions: LMSInitalize must be called at the beginning of a session; LMSGetValue can be used to retrieve information from the learning platform; LMSSetValue can be used to pass bookmarking and results data back to the platform; LMSCommit should be called to ensure that all data is stored on the platform permanently; and LMSFinish must be called at the end of the session.

Cross-domain scripting

Content developers should be aware that if the adapter and the content pages are hosted on different domains (for example, at www.platform.com/adapter.htm and www.publisher.com/content.htm), then for security reasons, modern browsers will not allow data to be passed between the two. There are a number of possible methods that learning platforms can use to get around this. The best practice is to consult with your potential platform partners on how they address the issue.

Tracking results data

Any publisher of interactive content faces the dilemma of what to do with results data. You may find that the most convenient and most powerful solution is to pass them to a third-party Learning Platform.

SCORM™ 1.2 supports three score fields: min, max and raw. Of these, min is of doubtful value; max is a useful indicator, for example, a student might have scored only 3/10 because he only completed three questions; and raw contains the main score. Under SCORM™ 1.2, this should always be passed as a percentage. SCORM™ 1.3 adds a fourth field: normal (short for normalised). The normal field should contain a percentage score while the other fields could contain raw marks.

Assume a unit worth a total of 10 marks and a student who has completed work worth 5 marks, scoring 3 out of the possible 5. These results could be passed using script along the following lines:

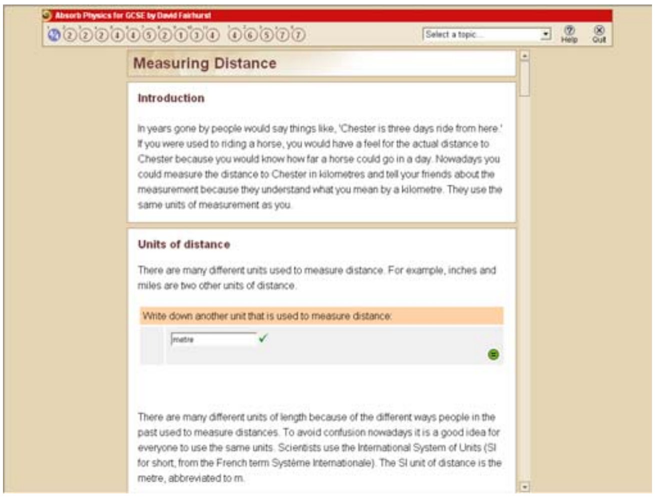
```
If (API != null) {
     API.LMSSetValue(“cmi.core.score.normal”, 30);
     API.LMSSetValue(“cmi.core.score.raw”, 3);
     API.LMSSetValue(“cmi.core.score.max”, 5);
}
```

More complex tracking of student abilities can be achieved by using the objectives fields. Refer to the SCORM™ documentation and check with your potential platform partners to see whether they provide meaningful support for objectives.

Supporting bookmarking

Bookmarking is a useful feature and allows learners to mark their place in the content if, for instance, they need to close down a session before completion of an object. SCORM™ provides two key fields to allow this: lesson_location and suspend_data. The differences are that suspend_data is intended to hold more extensive state information than lesson_location and the learning platform is meant to persist suspend_data between multiple attempts at the same unit.

Both fields can be encoded as you wish (your learning object is both producer and consumer of the data).



Measuring Distance

Introduction

In years gone by people would say things like, 'Chester is three days ride from here.' If you were used to riding a horse, you would have a feel for the actual distance to Chester because you would know how far a horse could go in a day. Nowadays you could measure the distance to Chester in kilometres and tell your friends about the measurement because they understand what you mean by a kilometre. They use the same units of measurement as you.

Units of distance

There are many different units used to measure distance. For example, inches and miles are two other units of distance.

Write down another unit that is used to measure distance:

metre ✓

There are many different units of length because of the different ways people in the past used to measure distances. To avoid confusion nowadays it is a good idea for everyone to use the same units. Scientists use the International System of Units (SI for short, from the French term Système Internationale). The SI unit of distance is the metre, abbreviated to m.

(Screenshot from Absorb courseware, by Crocodile Clips)

If, for example, the student leaves the above unit, having answered a single question as shown, the unit might write the following information to the learning platform before finishing:

```
if (API != null) {
      API.LMSSetValue(“cmi.lesson_location”, “#ccTop”);
      API.LMSSetValue(“cmi.suspend_data”, “q1~4~4~q1a1#metre#true”);
      API.LMSSetValue(“cmi.core.exit”, “suspend”);
}
```

In this implementation, #ccTop is a bookmark indicating that the student is still at the top of the page, while the text q1~4~4~q1a1#metre#true indicates that the student has scored 4 out of 4 for question q1, and the edit box called q1a1 has been completed with the word “metre” which has been marked correct. Note that the exit field is also set to “suspend” to confirm that the learning object wishes to bookmark itself.

The learning platform knows that the student has part-completed this particular unit and the next time it is launched, it can automatically reset itself to the same state that the student left it in, using script along the following lines:

```
if (API != null) {
      location = API.LMSGetValue(“cmi.lesson_location”);
      if (location != “”) { //the object has been bookmarked
            data = API.LMSGetValue(“cmi.suspend_data”);
            score = API.LMSGetValue(“cmi.core.score.raw”);
            //etc., using these fields to reset state in the object
      }
}
```

Finishing the session

As well as making sure that it has written up-to-date results data and checked whether bookmarking is required, it is desirable that a learning object should indicate the lesson’s status (for example, “completed”, “incomplete”, “not attempted”) and how the student exited (for example, “time-out”, “suspend” or “”, the last meaning normally). It will usually call LMSCommit to ensure that all data is stored permanently by the learning platform, and then is required to call LMSFinish, as shown in the following example:

```
If (API != null) {
     API.LMSSetValue(“cmi.lesson_status”, “completed”);
     API.LMSSetValue(“cmi.exit”, “”);
     API.LMSCommit(“”);
     API.LMSFinish(“”);
}
```

The learning object should try to ensure that this code is always called, even if the student closes the browser, for example, by pressing Alt-F4.

Having called API.LMSFinish, the learning object should leave all further housekeeping to the platform, which may wish to navigate immediately to another learning object without closing the current window.

Although it seems trivial, the call LMSFinish has a significant impact on the way in which learning content can be presented. Rather than relying on the student for navigation, the learning object can determine when it has finished and allow the learning platform to take control of navigation.

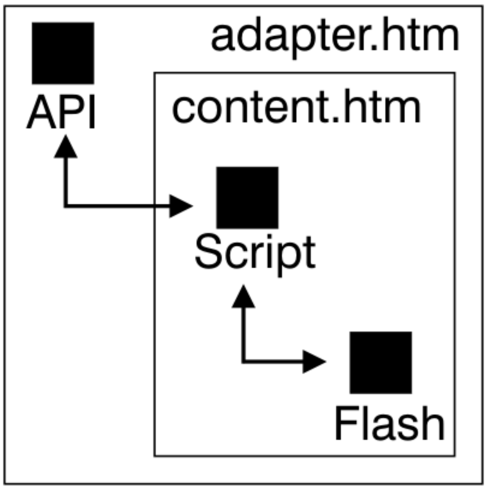
Error handling

For the sake of clarity, none of the code snippets in this section have included error handling. In practice it is important that learning objects show errors returned by the platform, as this is the quickest way to reveal misunderstandings between content and platform. Error handling can be implemented by calling code along the following lines after every other call:

```
errcode = API.LMSGetLastError();
if (errcode != 0) {
    alert(API.LMSGetErrorString(errcode));
}
```

Using the SCORM™ run-time with Flash animations

A considerable amount of learning content is now created using Macromedia Flash. This does not present any significant barrier to using the SCORM™ run-time. Data being written to the learning platform is passed out of the Flash animation using the Flash function fscommand, while bookmarking data can be passed into the animation by setting its parameters. Data then follows a two-stage journey between the Flash animation and the API, as illustrated in the figure below:



Other functions

This guide has illustrated some of the key benefits available to content authors implementing the SCORM™ run-time. Many other functions are also available: the learning object can store student preferences on the platform, AND return student comments or work requiring teacher assessment. As well as simple marks data, competency data can be tracked against learning objectives. When using the more advanced features of SCORM™, content developers should check to ensure that their platform partners use this data in meaningful ways.

Appendix 5

Examples

The following examples illustrate how packaging can be used.

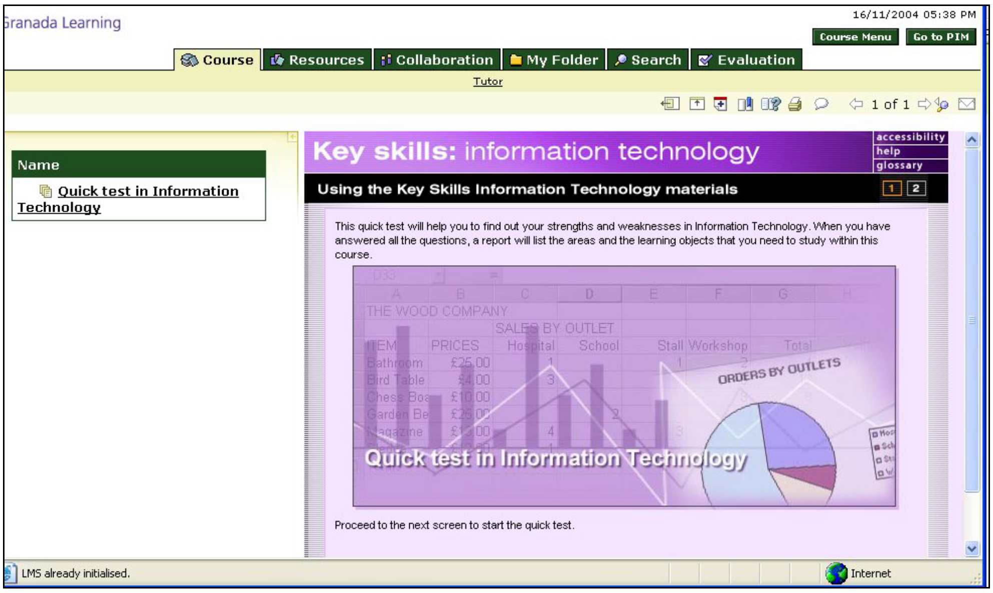
One learning object course – Organization extract

```
<organizations default="TOC1">
   <organization identifier="TOC1">
      <title>Key Skills IT Level 3</title>
        <item identifier = "ITEM33" identifierref = "RESOURCE177">
          <title>Quick test in Information Technology</title>
     </item>
   </organization>
</organizations>
```

and Resource extract

```
<resource identifier="RESOURCE177" adlcp:scormtype="sco" type="webcontent"
href="htm/welcome/02/quicktest_01.htm">
          <file href="htm/welcome/02/quicktest_01.htm"/>
          <file href="css/keyskills.css"/>
          <file href="css/question.css"/>
          <file href="images/content/welcome/02/intro.jpg"/>
          <file href="images/content/welcome/02/lo_title.gif"/>
          <file href="images/general/shim.gif"/>
          <file href="images/general/interface/keyskills_header.jpg"/>
          <file href="images/general/interface/top_gloss_hi.jpg"/>
          <file href="images/general/interface/top_gloss_lo.jpg"/>
          <file href="images/general/interface/top_help_hi.jpg"/>
          <file href="images/general/interface/top_help_lo.jpg"/>
          <file href="scripts/apiwrapper.js"/>
          <file href="scripts/general.js"/>
</resource>
```

VLE view



This quick test will help you to find out your strengths and weaknesses in Information Technology. When you have answered all the questions, a report will list the areas and the learning objects that you need to study within this course.

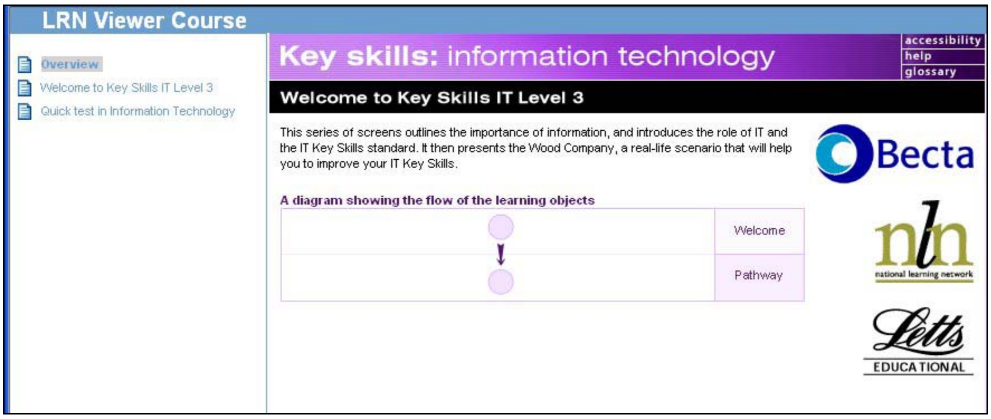
Proceed to the next screen to start the quick test.

Showing extracts from the Manifest for two common Aggregations as linear and hierarchical would make the relationship between the Manifest and the Root Aggregation and the Table of Content transparent. This could also be extended to show more complex Aggregations if required.

Learning object – Linear course example

```
<organizations default="TOC1">
  <organization identifier="TOC1" structure="linear">>
      <title>Key Skills IT Level 3</title>
    <item identifier="ITEM31" identifierref="RESOURCE172">
      <title>Overview</title>
    </item>
    <item identifier="ITEM32" identifierref="RESOURCE173">
      <title>Welcome to Key Skills IT Level 3</title>
    </item>
    <item identifier="ITEM33" identifierref="RESOURCE177">
      <title>Quick test in Information Technology</title>
    </item>
      </organization>
    </organizations>
```

VLE view



LRN Viewer Course

Overview
Welcome to Key Skills IT Level 3
Quick test in Information Technology
Key skills: information technology
accessibility
help
glossary

Welcome to Key Skills IT Level 3

This series of screens outlines the importance of information, and introduces the role of IT and the IT Key Skills standard. It then presents the Wood Company, a real-life scenario that will help you to improve your IT Key Skills.

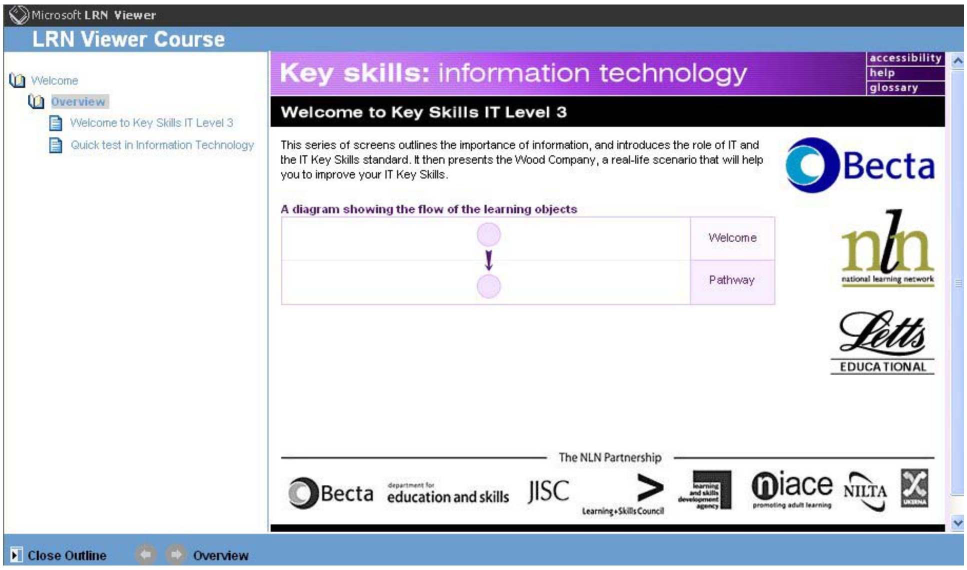
A diagram showing the flow of the learning objects

Welcome
Pathway
Becta
nln
national learning network
Letts
EDUCATIONAL

Learning object – Hierarchical course example

```
<organizations default="TOC1">
  <organization identifier="TOC1" structure="hierarchical">
     <title>Key Skills IT Level 3</title>
  <item identifier="it_v9-MANIFEST1_ITEM4" isvisible="1" parameters="">
     <title>Welcome</title>
  <item identifier="ITEM31" identifierref="RESOURCE172">
     <title>Overview</title>
  <item identifier="ITEM32" identifierref="RESOURCE173">
     <title>Welcome to Key Skills IT Level 3</title>
  </item>
  <item identifier="ITEM33" identifierref="RESOURCE177">
     <title>Quick test in Information Technology</title>
    </item>
  </item>
  </item>
 </organization>
</organizations>
```

VLE view



Welcome to Key Skills IT Level 3

This series of screens outlines the importance of information, and introduces the role of IT and the IT Key Skills standard. It then presents the Wood Company, a real-life scenario that will help you to improve your IT Key Skills.

A diagram showing the flow of the learning objects

(Examples kindly provided by Granada Learning)

Appendix 6

Standards and specifications

The following standards and specifications are included in SCORM™ 2004:

IEEE Data Model For Content Object To Learning Management System Communication (IEEE P1484.11.1 Draft)

IEEE ECMA Script Application Programming Interface for Content to Runtime Services Communication (IEEE 1484.11.2-2003)

IEEE Learning Object Metadata (LOM) (IEEE 1484.12.1-2002)

IEEE Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata (IEEE P1484.12.3 Draft)

IMS Content Packaging Version 1.1.3

IMS Simple Sequencing Version 1.0

Appendix 7

Bibliography

Becta Corporate Plan 2004-7

SCORM Best Practice Guide for Content Developers, 1st edition, February 2003, Carnegie Mellon

Virtual Learning Environments – Institutional Readiness, Becta

The Becta Learning Platform Conformance Regime Version 2.3

IMS Content Packaging Best Practice Guide Version 1.1.3 Final Specification (see also version 1.1.4)

NLN Round 4 Materials Development Programme Supplier's Handbook

Towards a Content Best Practice Guide for UK Compulsory Education, Draft 0.3, August 2004, Granada Learning, Pearson Education, Ramesys and RM

Guidelines for Authors of Learning Objects, Rachel S Smith, NMC: The New Media Consortium

Paving the Way, National Learning Network

Content Packaging, Sequencing and QTI: A Strategy for Curriculum Online, April 2003, Mike Collett, Schemeta

Acknowledgements

Lesley Mackenzie-Robb, Vantaggio Ltd (Author)

Wilbert Kraan, CETIS

Christine Lewis, South Yorkshire e-Learning Partnership

Richard Edwards, DfES

John Bell, Ufi

Mike Collett, Schemeta

Alan Teece, Granada Learning Group

Naomi Rich, Learn.co.uk

Diana Forster, OUP

Henry Ayres, Harcourt

Colin Young, Actis Ltd

Tom O’Leary, Public Records Office (National Archives)

John O’Donovan, BBC

Andrew Potter and Mike Edge, RM Learning Products

Crispin Weston, Alpha Learning

David Hassell, Michelle Morris, Barry Kruger and Luke Bennett, Becta

The author would also like to thank the Learning Platform Stakeholder Group for their contribution.