TECHNICAL REPORT

ISO/IEC TR 12785-3

First edition
2012-05-01

Information technology — Learning, education, and training — Content packaging —

Part 3: Best practice and implementation guide

Technologies de l'information — Apprentissage, éducation et formation — Paquetage du contenu —

Partie 3: Meilleure pratique et guide de mise en application

Reference number
ISO/IEC TR 12785-3:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword iv

0. Introduction v

1 Scope 1

2 Normative references 1

3 Terms and definitions 1

4 Abbreviated terms 4

5 Using content packages — Use cases and practices 4

5.1 Typical practice 4

5.2 Keeping control over resources after a package has been published 5

5.3 Aggregate content at an appropriate level of granularity 6

5.4 Specialized packages with combination of various specifications 8

5.5 Working with child-manifests and applying IMS simple sequencing rules using the new IPointer mechanism 9

5.6 Packaging METS and other complex object encodings 10

5.7 Using alternative organization structures, such as topic maps 12

5.8 Working with alternative resources 13

5.9 Avoiding repeated lists of the same assets for different resources 15

5.10 Working with local and global identifiers 17

5.11 Support for multiple languages in titles 17

Annex A (informative) **Document provenance** 19

Annex B (informative) **Intellectual property acknowledgements** 20

Bibliography 21

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 12785-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 36, *Information technology for learning, education and training*.

ISO/IEC 12785 consists of the following parts, under the general title *Information technology — Learning, education, and training — Content packaging*:

- *Part 1: Information model*

- *Part 2: XML binding*

- *Part 3: Best practice and implementation guide* [Technical Report]

0. Introduction

0.1 Purpose and overview

The primary focus of this part of ISO/IEC 12785 is on sharing existing best practice and providing suggested practice for implementing the functionality included in this part of ISO/IEC 12785. The ISO/IEC 12785 series has been commonly used in the learning, education, and training (LET) domain and this part of ISO/IEC 12785 is intended to support this use. It focuses on the construction of instances of manifest documents and the content packages they define.

0.2 Compatibility

Given the widespread adoption of IMS Content Packaging and the proliferation of hundreds of thousands of IMS content packages, it is important that existing software components continue to process content packages they were designed to handle, and that new software components conforming to the ISO/IEC 12785 series also process the older IMS content packages as designed. Newer systems will need the ability to process the new extension objects[1] introduced in the ISO/IEC 12785 series that enable linking and referencing behaviors. The functionality of these new extension objects are described in Clause 5 of this part of ISO/IEC 12785, and normative descriptions are contained in ISO/IEC 12785-1.

The new extension objects are defined in a separate namespace that leverages the extension points and semantics of the ISO/IEC 12785 series without affecting the existing IMS Content Packaging namespace. ISO/IEC 12785 also separates the lists of vocabulary terms used by certain objects in the information model (and a dedicated new namespace) from the model itself. These details are contained in the IMS GLC Specification Development Note 11: Vocabulary Definition, Registration, and Maintenance Procedures.

By taking this approach, we hope that the best of the past is preserved as it provides a strong foundation for future growth without having to alter the structural integrity of the ISO/IEC 12785-1 information model (a detailed, normative description of backwards and forwards compatibility is contained in ISO/IEC 12785-1).

[1] The extension refers to those elements that have been newly introduced with IMS Content Packaging version 1.2 which is a source of ISO/IEC 12785.

Information technology — Learning, education, and training — Content packaging

Part 3:
Best practice and implementation guide

1 Scope

This part of ISO/IEC 12785 presents use cases and shows how they are satisfied by ISO/IEC 12785-1 which is derived from the IMS Global Learning Consortium (GLC) Content Packaging version 1.2 (v1.2).

Though not exhaustive, the range of use cases presented in this part of ISO/IEC 12785 illustrate how the most common issues in the creation, management, and playback of learning material can be addressed by the ISO/IEC 12785 series. The use cases were contributed by various implementers and users of the IMS Content Packaging and are based on years of practice.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated reference, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 12785-1:2009, *Information technology — Learning, education, and training — Content packaging — Part 1: Information model*

IETF RFC 1951 (1996), *DEFLATE Compressed Data Format Specification version 1.3*

3 Terms and definitions

For purposes of this document, the following terms and definitions apply.

3.1
child manifest
complete, subordinate manifest contained in or referenced by the parent manifest

NOTE 1 According to IMS Content Packaging version 1.2 (and ISO/IEC 12785-1), a manifest can contain more than one child manifest.

NOTE 2 A manifest can include a reference to a child manifest that is external to the interchange package.

NOTE 3 A child manifest describes a complete logical package that is part of the larger logical package defined by its parent manifest.

NOTE 4 A child manifest can be local or remote.

Adapted from ISO/IEC 12785-1:2009.

3.2
content

individual file or multiple files usable in learning, education and training

NOTE 1 A logical unit of usable (and reusable) information can be described by a logical package.

NOTE 2 A logical package can contain one or more units of content.

3.3
content file

collection of files, including at least one manifest file, and conforming to the ISO/IEC 12785-1 information model and the ISO/IEC 12785-2 XML binding

NOTE Content files can be local or remote.

3.4
control file

single computer file that governs the binding of the Content Packaging Information Model (CPIM) to make it suitable for machine processing

NOTE A software component can refer to a control file when assessing the validity of a bound instance of the information model or to guide the creation of a bound instance of the information model. For example, a file containing an XML schema can be used as a control file for an XML binding of a manifest.

[ISO/IEC 12785-1:2009, 3.4]

3.5
interchange package

set of usable (reusable) LET content that is exchanged among computing systems used for information technology for learning, education and training (ITLET) purposes

NOTE An interchange package can be instantiated in a single compressed binary file (package interchange file) or as a collection of files on portable media (e.g. CD, DVD, USB memory device).

[ISO/IEC 12785-1:2009, 3.5]

3.6
manifest

description of files and any logical relationships between them, contained or referenced in a content package

3.7
metadata

⟨content packaging⟩ descriptive information about logical packages, logical organizations, content, and files

NOTE 1 Metadata can be assigned to any of the components within the logical package including the manifest.

NOTE 2 Any binding of a metadata object is permitted. Each object of metadata can be local or remote.

[ISO/IEC 12785-1:2009, 3.12]

3.8
namespace

XML namespace identified by a URI reference

NOTE Namespace in Content Packaging follows W3C recommendation, *Namespaces in XML 1.0 (Second Edition)*.

[ISO/IEC 12785-1:2009, 3.13]

3.9
organization
logical relationships, such as a hierarchical tree, among a unit of content

NOTE More than one logical organization can be described in a manifest.

3.10
package
unit of usable (and reusable) LET content

NOTE 1 This can be part of a learning course that has instructional relevance outside of a LET content aggregation and can be delivered independently, as an entire learning course or as a collection of learning courses.

NOTE 2 A package is able to stand-alone, that is, it contains all the information needed to use the contents for learning, education, and training when it has been unpacked.

[ISO/IEC 12785-1:2009, 3.14]

3.11
package reader
software that processes an interchange package by checking statements in the manifest against corresponding contents and organization

NOTE 1 A package reader can process both logical and physical packages.

NOTE 2 The term “process” may include the retrieval and storage of information referenced by the manifest, the decompression or unpacking of local files from a PIF, and the retrieval and/or logging of addresses of remote files

3.12
package writer
software that creates or modifies an instance of an interchange package and assembles content file(s) and other files declared local to the interchange package and writes them to the targeted interchange package binding, or delegates those tasks to another software typed process

[ISO/IEC 12785-1:2009, 3.17]

3.13
package interchange file
PIF
instantiation of an interchange package which is physically encapsulated as a compressed binary file conforming to IETF RFC 1951 (1996)

NOTE 1 An interchange package may be instantiated in a format other than a package interchange file (PIF).

NOTE 2 The representation (binding) is usually expressed in XML.

EXAMPLE An interchange package can be instantiated as a collection of files on removable media, e.g. CD, DVD, USB memory device, or compressed using another format such as .zip, .tar, .jar, .cab.

[ISO/IEC 12785-1:2009, 3.15]

3.14
resource (in content packaging)
one URL entry point and zero or more references to files that are required before the content is launched

NOTE The files described by a resource can be local or remote.

3.15
unit of content
either a file or a grouping of files which can be represented within a manifest

4 Abbreviated terms

CPIM	Content Packaging Information Model
ITLET	Information Technology for Learning, Education and Training
LET	Learning, Education, and Training
LMS	Learning Management System
LOM	Learning Object Metadata
METS	Metadata Encoding and Transport Schema
PIF	Package Interchange File
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
URI	Uniform Resource Identifier (IETF RFC 3986)
XML	Extensible Markup Language (W3C XML)
XSD	XML Schema Definition

5 Using content packages — Use cases and practices

The use cases in this clause illustrate key functionalities of ISO/IEC 12785 by focusing on particular goals that users of the standard may have and then outlining how such goals can be achieved. The section is not exhaustive neither in illustrating all features of the standard nor in outlining all uses of a specific feature.

5.1 Typical practice

Typically, a package contains all its resources in the PIF and uses a ZIP archive for its interchange format. It has only one manifest; no child-manifests are included. All content files are declared in the manifest and are included physically in the package. When material needs to be presented in more than one language or accessibility modality, two types of organization are contained in the manifest.

Content packages that comply with the most popular content-packaging profile, ISO/IEC 29163 SCORM, also typically follow this simple pattern.

Simple content packages are likely to be most widely supported in various Learning Management System (LMS), and therefore, are the most robust in interoperability.

Table 1 — Use case of the classic package

Level:	Primary use case
Actors:	Primary: Content authors[2] Secondary: Content repositories
Stakeholders:	Content developers
Interest:	Content-development tool developers
Basic flow of events:	• The content author creates required content. • A PIF (usually a ZIP file) is created that includes a manifest file and Web content resources simply using a packaging tool such as RELOAD[3]. • The PIF is deposited into a repository and/or LMS.
Alternative flow of events:	• A content author creates ISO/IEC 29163 SCORM compliant content. • A PIF is created including all ISO/IEC 29163 SCORM compliant files, such as metadata. • The PIF is deposited into a repository and/or LMS.
Success factors:	• All resources are contained within the PIF. • Packages are easily ingested by (legacy) compliant LMS systems.

Note: The “Simple_Manifest_Core” package in the set of examples that is included with this standard is a typical instance of a simple package. The example packages are available on the IMS website: http://www.imsglobal.org/content/packaging/index.html

5.2 Keeping control over resources after a package has been published

Originally, a PIF, such as a ZIP archive, was intended to function solely as a means of transporting a course from one delivery environment (e.g., an LMS) to another. In practice, however, often a course is delivered from the PIF after the PIF has been imported in an LMS, thereby changing the nature and use of the PIF that contains the course from a transport mechanism to a content management device.

Though this does work, there are cases where it is desirable not to copy the content that is aggregated in a PIF, but have it reside on a central server. That way, access to the resources can be controlled more easily; content files can be updated at any point in time and use monitored more accurately by the publisher.

[2] Content authors design the logical structure of content,e.g. teacher, professor, and instructional designer.

[3] See the reference: http://www.reload.ac.uk/tools.html.

Table 2 — Use case of keeping control over resources

Level:	Primary use case
Actors:	Primary: Content authors Secondary: Content repositories and learners
Stakeholders:	Content managers
Interest:	Content authoring tool developers
Basic flow of events:	1) The content author searches for content files on a variety of repositories. 2) The content author composes a content package manifest that contains absolute references to the remote files in the repositories. 3) The content package manifest is sent to a variety of LMSs. 4) A learner accesses the content package manifest in an LMS. 5) Content files are retrieved from the remote repositories independent from LMS.
Alternative flow of events:	1) The content author creates new content files and deposits them in one or more repositories. 2) The content author composes a content package manifest that contains absolute references to the resources located remotely in the repositories. 3) References in the content package manifest are coded to uniquely identify the instances of the manifest that are sent to a specific LMS or organization. 4) An LMS can be granted or denied access to the resources by the repositories, depending on the construction of the references or some other authentication and authorization mechanism determined by the repository.
Success factors:	• The content owner can control access to content files at any point in time. • The content owner can monitor content file usage at any point in time. • Updates to content files are propagated automatically.

5.3 Aggregate content at an appropriate level of granularity

A sharable content object (SCO) of ISO/IEC 29163 SCORM is the lowest aggregation level at which a learner’s progress through a course can be tracked in ISO/IEC 29163 SCORM. Despite this fact, SCOs often contain multiple pages of content. This makes it difficult to track precisely where a learner is in the material.

Child-manifests can be used to resolve this issue. Individual child-manifests can describe individual sub-components (e.g. one manifest for each individual page of content) that would otherwise be undifferentiated in a single SCO. As defined above (3.1) a child-manifest is a complete, subordinate manifest contained within or (from ISO/IEC 12785 onwards) referenced by a main or parent manifest. A child-manifest describes a complete logical package that is part of the larger logical package defined by its parent manifest.

Table 3 — Use case of aggregation content

Level:	Primary use case
Actors:	Primary: Content authors Secondary: Content repositories
Stakeholders:	Stakeholder: Content developers
Interest:	Content development tool developers
Preconditions:	• All basic content files have already been authored. • A hierarchy of content packages has been created; the package and manifest on the highest level in this hierarchy references child manifests that correspond to more specific contents on a lower level. • The content development tool supports child-manifests, either contained within or referenced by the parent manifests.
Basic flow of events:	1) Using a content development tool, a content author gathers basic content into what will be independently reusable content packages. 2) The content author turns each of these into a content package which consist of single-page SCOs. 3) The content author aggregates the packages into a single logical package in one PIF.
Postconditions:	• The PIF contains one manifest document that contains the child-manifests of the aggregated packages (see examples, below). • Alternatively, the PIF contains a parent manifest that references child-manifests that may be located in subdirectories defined in the content package (see examples, below).
Success factors:	• The learner’s progress can be tracked at an appropriate level of granularity. • Content assets can be authored and managed at a aggregation level that makes reuse as easy as possible.

Child-manifest examples:

As an example of the use of child-manifests, suppose a content author was developing a lesson on camping based on three existing content packages: “How to construct a tent”, “How to start a fire”, and “How to make campfire snacks”. The content author can either (a) include the manifests for these existing content packages in a parent manifest, or (b) reference child-manifests from the parent manifest. Referenced child-manifests may be local to the interchange package or may be hosted remotely (for example in a repository).

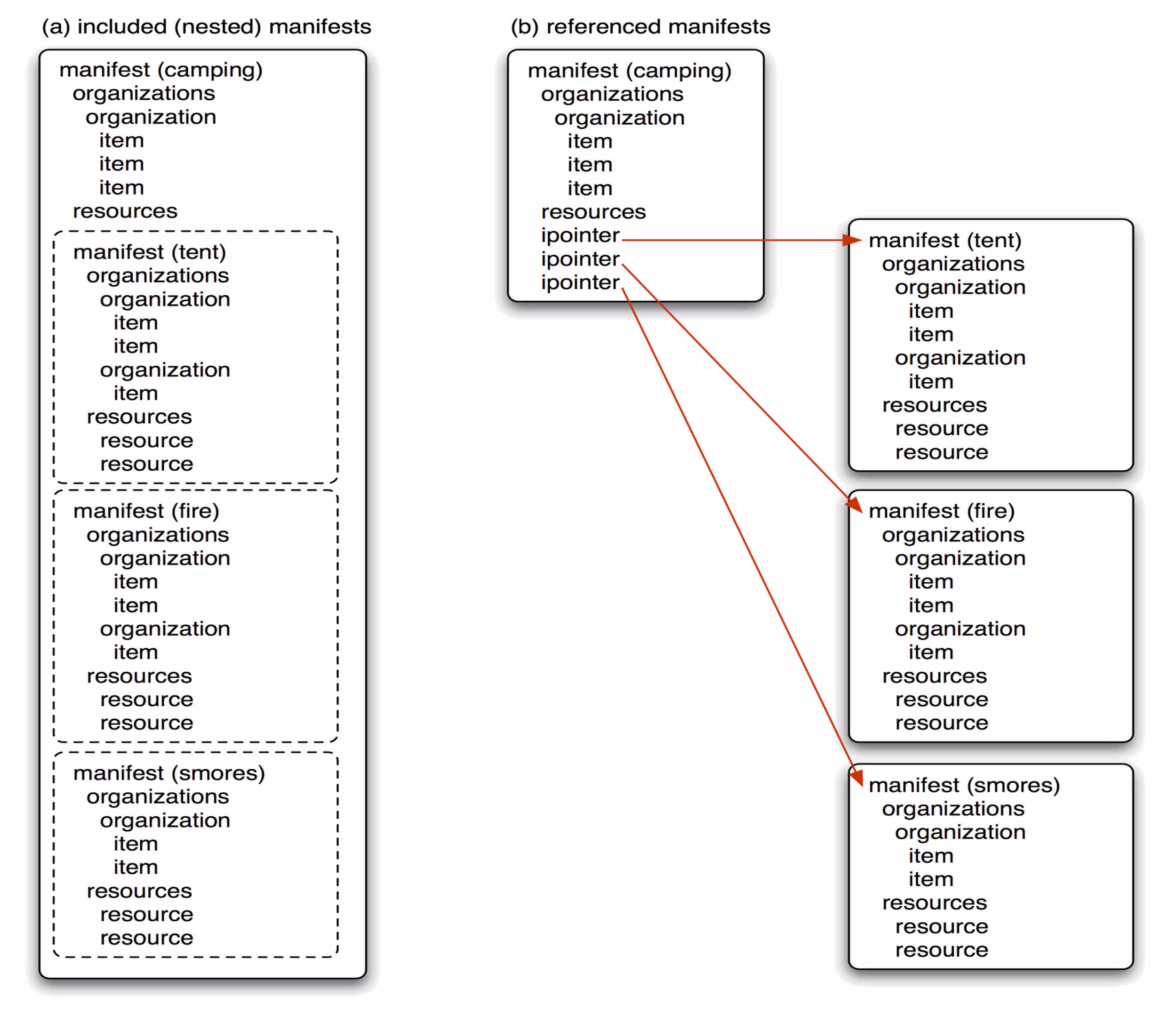


Figure 1 — An example of the use of child-manifests

5.4 Specialized packages with combination of various specifications

This section discusses packages which contain XML data as local content. These packages are considered specialized because this configuration is often used in conjunction with other IMS specifications including:

- IMS ePortfolio
- IMS Learning Design
- IMS Learner Information Package
- IMS Question & Test Interoperability
- IMS Simple Sequencing

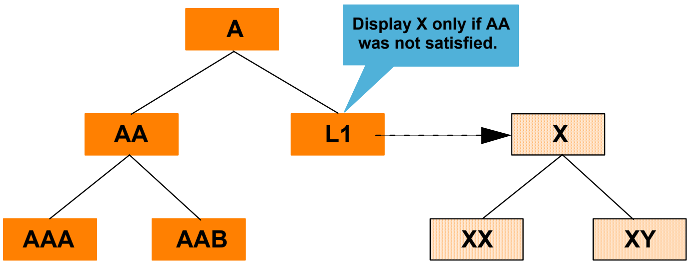
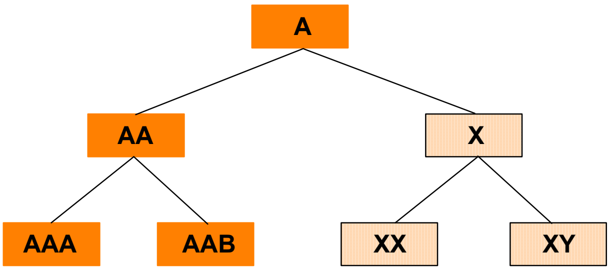
Guidance on how each of these specifications uses content packaging is provided in the relevant IMS documentation set[4].

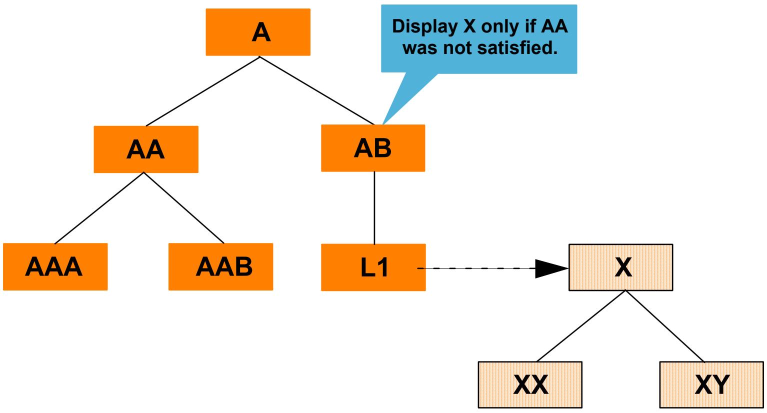
[4] IMS specifications can be found in: http://www.imsglobal.org/specifications.html

5.5 Working with child-manifests and applying IMS simple sequencing rules using the new IPointer mechanism

Some specifications, such as IMS Simple Sequencing, add their own XML attributes to Item structure of ISO/IEC 12785. Thought needs to be given as to how the information conveyed by those attributes affect the tree-like structures that can be built with Item objects in an Organization objects. Specifically, conflicts may exist among several sets of sequencing rules when a parent Manifest declares an association with one or more child Manifest objects and all contain sequencing instructions. This is conceptually no different from cases where sequencing rules exist on a parent Item that contains at least one child Item that also has sequencing rules. When using IPointer objects, a key point to bear in mind is whether an Item becomes a parent to a target Item in the external Manifest to which the IPointer points. This same logic applies to Manifest objects that contain internal references to child Manifest objects.

Table 4 — Use case of child-manifests and sequencing rules

Level:	Primary use case
Actors:	Primary: Content authors and instructional designers Secondary: Content tool developers
Stakeholders:	Content developers
Preconditions:	- A requirement exists to construct a new content package with remediation (i.e. an exercise for students requiring extra practice). - The remediation should only be presented to the learner if the learner fails a quiz. Conceptually, the new design may look like Figure 2, where A represents the new content package, AA contains the quiz and X is the content package to be used for remediation, if the learner does not satisfy AA.  A Display X only if AA was not satisfied. AA L1 X AAA AAB XX XY **Figure 2 — Conceptual design of a new package** - The remote package (i.e., X, in Figure 2) cannot be altered by the content package author. - Conceptually, if a package reader were to process the structure in Figure 2, the resulting structure should look like that in Figure 3.  **Figure 3 — Conceptual design of processed structure**

Basic flow of events:	1) An instructional designer searches a repository and finds a content package that has suitable remediation. The instructional designer needs to apply a sequencing rule to X that says it should be displayed only if AA was not satisfied. 2) However, L1 is a linking object that will be completely replaced and cannot have associated sequencing rules to node A. Instead, the instructional designer adds a new Item that will be the parent of the external structure. 3) The instructional designer applies sequencing rule for X to the new parent (AB). Conceptually, the new content package would look like that in Figure 4. In order to make the situation shown in Figure 4 work, the instructional designer needs to introduce global variables. For AB to know if AA was satisfied, the satisfaction status of AA would need to be mapped to such a global variable, which can be read by AB.  A Display X only if AA was not satisfied. AA AB AAA AAB L1 X XX XY **Figure 4 — Conceptual design of the applied sequencing rule**
Success factors:	• References to external manifests are processed and displayed as the instructional designer originally intended, depending on learner responses. • Sequenced content is easily reused for a variety of instructional purposes.

5.6 Packaging METS and other complex object encodings

Several formats exist for aggregating complex digital objects in addition to ISO/IEC 12785. In the library community, for example, Metadata Encoding and Transport Schema (METS) is widely used, while ISO/IEC 21000-2 is used in many multimedia applications.

Ideally, complex digital objects should be transformable from one such format to another. However, this is not always possible or even desirable. Sometimes, it may be preferable to exchange a complex digital object in one format wrapped inside another. This approach can be used to exchange a complex digital object without affecting its native format, structural integrity, or semantic fidelity.

Table 5 — Use case of Packaging METS

Level:	Primary use case
Actors:	Primary: Content managers Secondary: Content repositories
Stakeholders:	Copyright holders
Interest:	Content authoring tool developers
Preconditions:	• Repository 1 cannot export objects as content packages. • The authoring tool and repository 2 are not designed to process any other types of complex digital objects other than content packages.
Basic flow of events:	1) A complex digital object in METS format is exported from repository 1. 2) The content manager imports the METS object into a content packaging tool as a resource. 3) The content manager checks the IMS resource-type vocabulary for the entry for METS objects and uses the content packaging tool to add the correct value in the Type characteristic of the Resource object. 4) The content manager adds appropriate metadata to a Metadata object to indicate the nature of the METS object. 5) The newly created content package is imported into repository 2.
Alternative flow of events:	1) A complex digital object in format A is exported from repository 1. 2) The content manager imports the format A object into a content packaging tool as a resource. 3) The content manager checks whether the IMS resource-type vocabulary has an entry for format A and finds that it does not. 4) Using the "IMS Vocabulary Definition, Registration & Maintenance Procedures", the content manager composes a value and proposes it to the format A community for submission to the IMS resource-type vocabulary. 5) The content manager uses the content packaging tool to add the new value in the Type characteristic of the Resource object. 6) The content manager adds appropriate metadata to a Metadata object to indicate the nature of the format A object. 7) The newly created content package is imported into repository 2.
Success factors:	• The structural integrity of the non-standard content packaging complex digital object is preserved in the exchange between repository 1 and 2. • Non-standard content packaging complex digital objects are successfully processed by tools that have been designed to process ISO/IEC 12785, only.

5.7 Using alternative organization structures, such as topic maps

Basically, ISO/IEC 12785 provides one default Structure type: “hierarchical”. “Hierarchical” is intended to support table-of-contents type structures to learning resources. However, many implementers have use cases for other means of organizing resources that are not defined by ISO/IEC 12785. Typically, these cases are driven by a need to provide richer or more complex learning activities. Examples of alternative organizations are directed graphs, flat lists, and standardized structures, such as ISO 13250-2 Topic Maps – Data Model.

Table 6 — Use case of alternative organization structures

Level:	Primary use case
Actors:	Primary: Content authors Secondary: Learners
Stakeholders:	Content authoring tool developers
Interest:	Content repository tool developers
Preconditions:	• A content author wants to structure learning resources using structures other than the one defined by ISO/IEC 12785. • The content has to function in an environment where not all software tools will support the desired alternative organization structures.
Basic flow of events:	1) The content author aggregates content files for an content package. 2) Using a specialized authoring tool, the content author creates an alternative organization structure in a separate document. 3) Using a standard content packaging authoring tool, the content author defines both the aggregated content files and the alternative organization structure as resources and creates a conventional table-of-contents style organization. 4) The content author makes the content package available in a suitable system, such as a repository.
Alternative flow of events I:	1) The content author aggregates content files for a content package. 2) Using a specialized authoring tool, the content author creates an alternative organization structure in a separate document. The alternative organization has direct references to the content files. 3) Using a standard content packaging authoring tool, the content author inserts the alternative organization structure under its own namespace in a separate Organization object within an Organizations object in a Manifest. 4) The content author also creates a conventional table-of-contents style Organization and concurrent Resources objects. The conventional Organization object is defined as the default. 5) The content author makes the content package available in a suitable system, such as a repository.

Alternative flow of events II:	1) The content author aggregates content files for a content package. 2) Using an authoring tool that implements both ISO/IEC 12875 and an alternative organization structure, the content author creations an alternative organization structure with an Organization object in the Manifest. The alternative Organization references resources in the Resources object. 3) Using the same authoring tool, the content author also creates a conventional table-of-contents style Organization and concurrent Resources objects. The conventional Organization object is defined as the default. 4) The content author makes the content package available in a suitable system, such as a repository.
Success factors:	• The content packages with alternative organization structures continue to function as conventional content packages for conventional content packaging readers and repositories. Systems designed to process alternative organization structures locate those structures and make them available to learners in addition to, or instead of, the conventional organization structures.

5.8 Working with alternative resources

Because the access requirements of different learners vary, the accessibility properties of educational content needs to be described such that a package processing tool can make a good decision about which content to render for a particular learner. Content that is described in this way can reside in more than one place. What this means is that, at or close to run time, a learning resource or component may be replaced or supplemented. The replacement or supplementation may relate to a resource or to a file or to files that form part of the resource. When working with a resource, accessibility metadata about that resource will commonly describe its properties including the location of any replacement or supplementary resources. In ISO/IEC 24751-3, each alternative resource that is associated with a primary resource can be found at the end of a URI that can point to a location internal to the system or external to it.

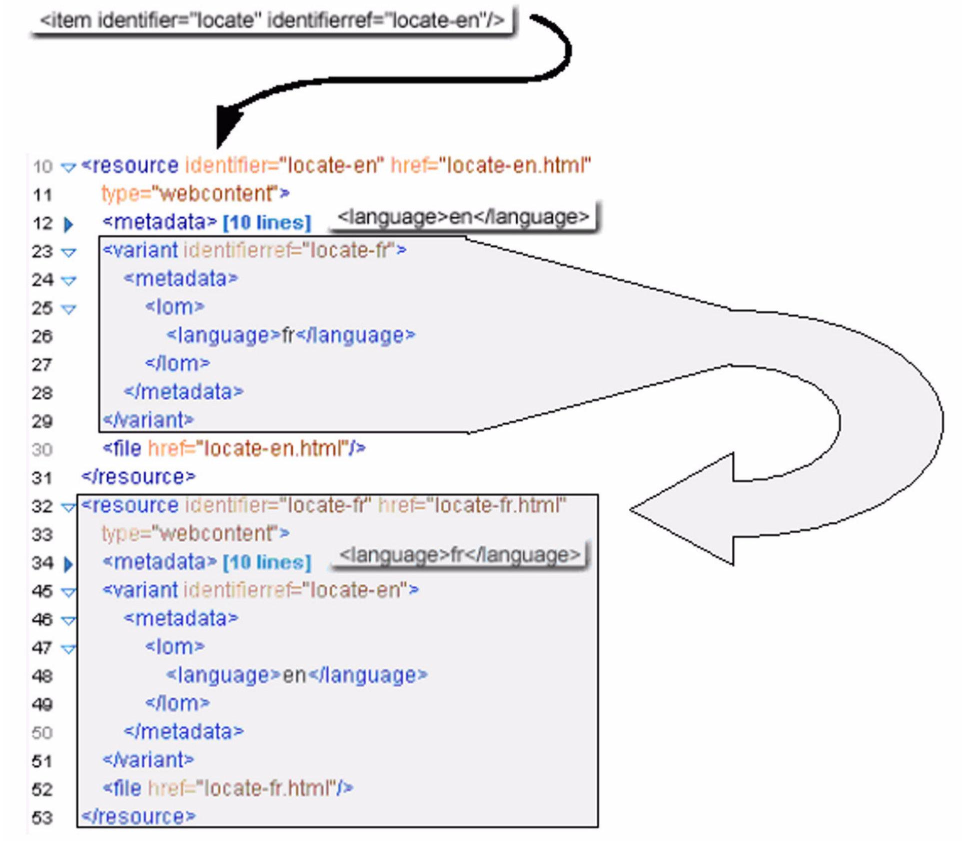
The mechanism provided is that a Resource object can have one or more associated variant Resource objects. Association is via a dedicated Variant object within a Resource that references the alternative Resource within the same Manifest. Where the alternative resource is external, it is referenced indirectly via the internal associated resource. The relationship is logically equivalent to containment of the variant resource within the one for which it is a variant.

ISO/IEC 12785 places no constraints on how the mechanism is used to structure resources, and several different ways to do this are possible, some of which are shown in the examples.

Table 7 — Use case of alternative resources

Level:	Primary use case
Actors:	Primary: Content authors Secondary: LMSs and learners
Stakeholders:	Content authoring tool developers
Interest:	Content repository and LMS tool developers
Preconditions:	• All distributed resource alternatives have been pulled into the Manifest when the resource set was packaged, so that the package is complete and the content stands alone. • The LMS has access to the learner's accessibility preferences. • The LMS can parse and process accessibility metadata.

Basic flow of events:	1) The content author gathers course material that is available in multiple languages, in this case English and French as shown in Figure 5. 2) Using an authoring tool, the content author describes the English resource illustrated in lines 10 – 31 of Figure 5. Because the resource is referenced directly from an Item object, it is the default resource. 3) The content author creates an alternative for this resource in French (described in lines 32 – 53). 4) The content author creates a Variant object to reference the French Resource (lines 23 – 29). 5) The content author creates the required Metadata object that describes the context of use of the alternative resource; in this case it says that the alternative is the French version. 6) On completion, the content author submits the package to a repository. 7) On playback, the LMS compares language accessibility preferences of the learner (i.e. French preferred) to the accessibility metadata in the package. 8) The LMS presents the variant resource to the Learner.
Alternative flow of events:	1) The content author gathers course material that is available in multiple languages; in this case English and French. 2) Using an authoring tool, the content author describes the English resource in one Manifest of one package. 3) The content author creates an alternative for this resource in French in a Manifest in a different package. 4) The content author creates a Variant object to reference an IPointer object that identifies the French Resource in the external package. 5) The content author creates the required Metadata object for the Variant object that describes the context of use of the alternative; in this case it says that the alternative is the French version. 6) In the package with the French Resource, the content author creates a Variant object to reference an IPointer object that identifies the English Resource in the other package. 7) The content author creates the required Metadata object for the alternative Resource object that describes the context of use of the alternative; in this case it says that the alternative is the English version. 8) On completion, the content author submits both packages to a repository. 9) On playback, the LMS compares language accessibility preferences of the learner (i.e. French preferred) to the accessibility metadata in the package. 10) The LMS processes the external package indicated by the IPointer object, and presents the desired variant resource to the learner.
Success factors:	• Alternative resources are supported (e.g., a version of a resource with modified display characteristics to support vision-impaired users). • Specific vocabulary terms for accessibility resources are supported for the Type characteristic of the Resource object.



```
<resource identifier="locate-en" href="locate-en.html"
  type="webcontent">
  <metadata> [10 lines]
  <variant identifierref="locate-fr">
    <metadata>
      <lom>
        <language>fr</language>
      </lom>
    </metadata>
  </variant>
  <file href="locate-en.html"/>
</resource>
<resource identifier="locate-fr" href="locate-fr.html"
  type="webcontent">
  <metadata> [10 lines]
  <variant identifierref="locate-en">
    <metadata>
      <lom>
        <language>en</language>
      </lom>
    </metadata>
  </variant>
  <file href="locate-fr.html"/>
</resource>
```

Figure 5 — A simple example for specifying choice of languages

Variant references may be from any Resource object to any peer Resource object (child of the same parent). A closer look at this example reveals that in fact the example contains not only the reference shown in Figure 5, but also the reverse reference, which may be a useful mechanism in some scenarios.

Any Resource can contain zero or more such references. This enables scenarios, such as

- A Resource has many Variant objects referenced within it.
- A referenced Variant (the alternative resource) has its own Variant objects.
- Variant objects could be hierarchically structured.
- A set of Resource objects, such as might be described with RDF, could have arbitrarily-structured Variant references.

5.9 Avoiding repeated lists of the same assets for different resources

Many users and content vendors prefer that learning resources have a similar look and feel. This is easily achieved by re-using the same images and styling information for all resources in a content package. However, because Resource objects are required to list all the content files that a resource uses, this approach has the disadvantage that it can lead to a lot of repetition in the Resources section of the Manifest. Therefore, ISO/IEC 12785 has a feature that lets content authors list commonly used content files once and reference that list from many different Resource objects.

Table 8 — Use case of avoiding repeated lists of the same assets

Level:	Primary use case
Actors:	Primary: Content authors Secondary: ISO/IEC 12785–compliant content-aggregation tools
Stakeholders:	Content authoring tool developers and LMS developers
Preconditions:	• For a significant number of LMSs targeted by the content author, dereferencing internal references and integrating the structures they point to into local Resource structures must be less costly than the storing and parsing repetitive XML.
Basic flow of events:	1) The content author creates Web-page resources for an content package in a single style, making use of a single set of recurring images, CSS style sheets, and JavaScript files. 2) When the content author imports the content files into the content-aggregation tool, the tool recognizes the repetition of the same set of files in multiple resources. 3) The content-aggregation tool groups the repeatedly used set of content files into one Resource object, which is not referenced by any Item object, and references that Resource from any other Resource objects that make use of the set via the Dependency object. 4) The LMS dereferences the Dependency references to the Resource that holds the repeatedly used set of files every time such a Dependency is declared in other Resource objects. 5) The LMS integrates the content files declared by the local Resource and the Resource referred to by the Dependency into a single representation for the learner.
Alternative flow of events:	1) The content author creates Web-page resources in a consistent style in an LMS. 2) On completion, the authoring tool rationalizes the number of content files required to render the resources down to one copy of each. 3) The content-aggregation tool groups those files that are used repeatedly into one Resource object, which is not referenced by any Item object, and references that Resource from any other Resource objects that make use of the set via the Dependency object. 4) The LMS dereferences the Dependency references to the Resource that holds the repeatedly used set of files every time such a Dependency is declared in other resources. 5) The LMS integrates the content files declared by the local Resource and the Resource referred to by the Dependency into a single representation for the learner.
Success factors:	• All content files are declared only once in a single Manifest. • The content files that are aggregated into one Resource are successfully marshaled at rendering time. • Disk space, processing cycles, and processing time are saved compared to declaring content files repeatedly.

5.10 Working with local and global identifiers

ISO/IEC 12785 uses the Identifier object to identify parts of a manifest. This local-identifier feature is used primarily to associate Item objects with Resources. Because the Identifier attribute is defined by the W3C's XML Schema specification as unique within each document, many tools can easily ensure that objects in a Manifest are unique and that references to them are unambiguous. The cost of this useful feature is that the form of the local identifier has a set of constrains. Most notably, these constraints preclude the use of many common forms of global identifiers, such as Uniform Resource Names (URNs). As a consequence, the Identifier object is not a good way to store globally unique identifiers for a logical package or any part thereof.

A much better way to store such global identifiers is in the Metadata object. Unlike the structural role of a local identifier, a globally unique identifier can be seen as a type of metadata; it conveys information about the content package, but does not play a role in the package itself. In addition, Metadata can easily accommodate the wide variety of globally unique identifier syntaxes, semantics, and data models that different communities use. Moreover, Metadata can contain an unlimited number of globally unique identifier instances in almost any form.

Table 9 — Use case of local and global identifiers

Level:	Primary use case
Actors:	Primary: Content authors Secondary: ISO/IEC 12785-compliant content-aggregation tools, content managers, and repositories
Stakeholders:	Content authoring tool, LMS, and content repository tool developers
Preconditions:	• The content author's community has defined a Learning Object Metadata (LOM) profile that mandates the use of globally unique identifiers of the “handle” type to all content packages.
Basic flow of events:	1) The content author aggregates a number of content files using a content-aggregation tool. 2) When the content author imports the content files into the content-aggregation tool, the tool assigns locally unique identifiers to the Item and Resource objects, and records those in ID attributes. 3) The content author uses the content-aggregation tool to author a LOM record that describes the new content package. 4) The content author enters a “handle” identifier in the LOM record. 5) While importing, the repository parses the LOM record and exposes the 'handle' identifier in its search and harvesting interfaces.
Success factors:	• Local identifier references are unambiguous and robust. • Any type or multiplicity of globally unique identifiers can be associated with the content package or a part thereof.

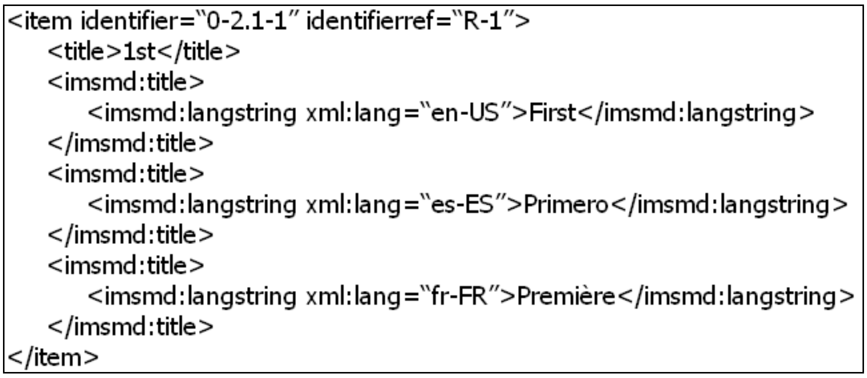
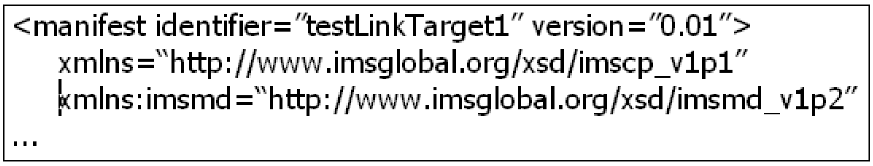
5.11 Support for multiple languages in titles

The Title object defined for Organization and Item object is just a label. It does not specify a language. Nor can the Title be repeated to accommodate multiple languages. Yet learning material that is intended to be used in multilingual environments or material used in language teaching requires the use of titles in more than one language or at least an indication of what language a title is in.

Language-sensitive titles can be added into a Manifest as extensions from another namespace as children of an Organization or Item object. For example, if one wanted to express representations of the same title in multiple languages, one could use the title element from the LOM. No restrictions exist on how many instances of LOM’s title element are included in an Item, Manifest, or Organization object, and LOM’s title element also has a means of indicating what language the title is in.

Table 10 — Use case of support for multiple languages

Level:	Primary use case
Actors:	Primary: Content authors and content translators Secondary: Content developers and content repositories
Stakeholders:	LMSs
Interest:	Content tool developers
Preconditions:	• The content package is intended to be used by English, French, and Spanish speaking learners. • LMSs are aware of the language preferences of their users. • LMSs can parse the LOM title element.

Basic flow of events:	1) A content author creates a content package English.
	2) The package is passed to a content translator to be translated into French.
	3) The translator opens the package, and adds the title in French, and inserts the French title into an IMS Metadata 'title' tag, with an xml:lang attribute value of “fr-FR”, as shown in Figure 6.
	
	`<item identifier=“0-2.1-1” identifierref=“R-1”>`
	`    <title>1st</title>`
	`    <imsmd:title>`
	`        <imsmd:langstring xml:lang=“en-US”>First</imsmd:langstring>`
	`    </imsmd:title>`
	`    <imsmd:title>`
	`        <imsmd:langstring xml:lang=“es-ES”>Primero</imsmd:langstring>`
	`    </imsmd:title>`
	`    <imsmd:title>`
	`        <imsmd:langstring xml:lang=“fr-FR”>Première</imsmd:langstring>`
	`    </imsmd:title>`
	`</item>`
	Figure 6 — An Item with a default Title in English, and supplementary titles in French and Spanish.
	4) The translator declares the IMS Metadata namespace identifier and namespace name in the Manifest, typically as part of the attributes for the root Manifest object as illustrated in Figure 7.
	
	`<manifest identifier=“testLinkTarget1” version=“0.01”>`
	`    xmlns=“http://www.imsglobal.org/xsd/imscp_v1p1”`
	`    xmlns:imsmd=“http://www.imsglobal.org/xsd/imsmd_v1p2”`
	`…`
	Figure 7 — An IMS Metadata namespace declaration in the Manifest element.
	5) The content package is then translated into Spanish, which repeats steps 3, 4, and 5 above to add titles in Spanish.
	6) The LMS compares the language preference of the learner (Spanish) and renders the package’s titles in Spanish.

Success factors:	• Learners see titles in a language that most closely matches their preferences.

Annex A
(informative)

Document provenance

Title	IMS Content Packaging Best Practice and Implementation Guide
Editors	Colin Smythe (IMS), Boyd Nielsen (Independent)
Co-chairs	Wilbert Kraan (JISC/CETIS), Jan Poston Day (Blackboard), Nigel Ward (DEST)
Version	v1.2 (CM/DN Draft v2.0)
Version Date	01 March 2007
Status	Contributing Member / Developers Network Draft v2.0
Summary	This document describes the IMS Content Packaging v1.2 Information Model objects, their relationships, and purpose. It also defines selected behaviors of software components that create or process IMS content packages and IMS manifest documents.
Revision Information	21 February 2007
Purpose	This document is circulated for CM/DN adoption. This document should be used to introduce the IMS Content Packaging v1.2 specification. Organizations are encouraged to implement this version of the specification and to provide feedback on their experience.
Document Location	Content Packaging Document Library.

Annex B
(informative)

Intellectual property acknowledgements

“W3C is a trademark (registered in numerous countries) of the World Wide Web Consortium; marks of W3C are registered and held by its host institutions MIT, ERCIM, and Keio.

“SCORM” is a registered trademark of the Advanced Distributed Learning Initiative, Office of the Deputy Under Secretary of Defense (Readiness), Readiness and Training, 1E525, 4000 Defense Pentagon, Washington, D.C. 20301.

Bibliography

[1] *IMS AccessForAll Meta-data Information Model Version 1.0 Final Specification*, J.Treviranus, A.Roberts, IMS GLC, Jul 2004

[2] IMS GLC Content Packaging Information Model Version 1.2 CM/DN Draft v2.0, C.Smythe, B.Nielsen, W.Kraan, J.Posten Day and N.Ward, IMS GLC, Inc., March 2007

[3] IMS GLC Content Packaging XML Binding Version 1.2 CM/DN Draft 2.0, B. Nielsen, W. Kraan, J.Posten Day and C.Smythe, N.Ward, IMS GLC, Inc., March 2007

[4] IMS GLC Content Packaging Best Practices and Implementation Guide Version 1.2 CM/DN Draft 2.0, B. Nielsen, W. Kraan, J.Posten Day and C.Smythe, N.Ward , IMS GLC, Inc., March 2007

[5] IMS GLC Content Packaging Primer Version 1.2 CM/DN Draft v2.0, B.Nielsen, W.Kraan, N.Ward, IMS GLC, Inc., March 2007

[6] IMS Content Packaging Information Mode v1.1.4, C.Smythe, A.Jackl, IMS GLC, Inc., October 2004.

[7] IMS Content Packaging XML Binding v1.1.4, C.Smythe, A.Jackl, IMS GLC, Inc., October 2004.

[8] IMS Learning Design Information Model version 1.0 Final Specification, R.Koper, B.Olivier, T.Anderson, IMS GLC, 2003

[9] IMS Learning Design Best Practice and Implementation Guide version 1.0 Final Specification, R. Koper, B. Olivier, T. Anderson, IMS GLC, 2003

[10] IMS Learner Information Packaging Information Model version 1.0 Final Specification, C.Smythe, F.Tansey, R.Robson, IMS GLC, 2001

[11] Metatdata Encoding & Transmission Standard(METS) Profile Schema 2.0, April 16, 2010

[12] IMS Question and Test Interoperability Information Model – Version 2.0 Final Specification," S.Lay, IMS GLC, 2005

[13] IMS Question and Test Interoperability XML Binding – Version 2.0 Final Specification, S.Lay, IMS GLC, 2005

[14] IMS Question and Test Interoperability Integration Guide – Version 2.0 Final Specification, S.Lay, IMS GLC, 2005

[15] IMS Question and Test Interoperability Meta-data and Usage Data – Version 2.0 Final Specification," S.Lay, IMS GLC, 2005

[16] "RDF/XML Syntax Specification (Revised)," W3C Recommendation 10 February 2004

[17] IMS Vocabulary Definition Exchange Best Practice and Implementation Guide, Version 1.0 Final Specification, A. Cooper, IMS GLC, Inc., 2005.

[18] IMS Simple Sequencing Best Practice and Implementation Guide — Version 1.0 Final Specification," M.Norton, C.Ostyn, A.Panar, B.Towle, IMS GLC, 2003

[19] W3C RDF/XML Syntax Specification (Revised) (2004)

[20] W3C XML Schema Part 2: Datatypes Second Edition (2004)

[21] W3C Recommendation, Namespaces in XML 1.0 (Second Edition), 16 August 2006

[22] ISO/IEC 24751-3:2008 Information Technology — Individualized adaptability and accessibility in e-learning, education and training — Part 3: " Access for all" digital resource description

[23] IEEE 1484.12.1-2002, IEEE Standard for Learning Object Metadata

[24] IETF RFC 3986 (2005), Uniform Resource Identifier (URI): Generic Syntax

[25] ISO/IEC 29163-1:2009 Information technology — Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition — Part 1: Overview Version 1.1

[26] ISO/IEC 29163-2:2009 Information Technology — Sharable Content Object Reference Model (SCORM) 2004 3rd Edition — Part 2: Content Aggregation Model Version 1.1

[27] ISO/IEC 29163-3:2009 Information technology — Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition — Part 3: Run-Time Environment Version 1.1

[28] ISO/IEC 29163-4:2009 Information technology — Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition — Part 4: Sequencing and Navigation Version 1.1

[29] ISO/IEC 13250-2:2006 Information technology — Topic Maps — Part 2: Data Model

[30] ISO/IEC 21000-2:2005 Information technology — Multimedia framework — Part 2: Digital Item Declaration

ICS 35.240.99

Price based on 22 pages