



中华人民共和国国家标准

GB/T XXXXX—XXXX

信息技术 学习、教育和培训 学习资源通用包装

Information technology - Learning, education and training -

Common cartridge for learning resources

(报批稿)

XXXX - XX - XX 发布

XXXX - XX - XX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前 言.....	II
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 缩略语.....	4
5 通用包装概念模型.....	4
5.1 通用包装概念图.....	4
5.2 概念模型描述.....	5
6 通用包装信息模型.....	8
6.1 通用包装信息模型结构.....	8
6.2 信息模型组件.....	9
7 内容包装.....	27
7.1 内容清单.....	27
7.2 文件夹内容类型.....	28
7.3 学习应用对象.....	30
7.4 Web 内容类型.....	30
7.5 包级 Web 内容类型.....	30
7.6 关联内容类型.....	31
7.7 讨论主题内容类型.....	31
7.8 Web 链接（URL）内容类型.....	31
7.9 评估内容类型.....	31
7.10 题库内容类型.....	32
7.11 通用包装包权限.....	32
7.12 Web 内容资源的路径.....	32
参考文献.....	36

前 言

本标准按照GB/T 1.1-2009给出的规则起草。

本标准由全国信息技术标准化技术委员会（SAC/TC 28）提出并归口。

本标准起草单位：北京邮电大学、华东师范大学、中国电子技术标准化研究院。

本标准主要起草人：李青、吴永和、祝智庭、余云涛、刘勇、李莹、辛雨菡、赵欢欢。

信息技术 学习、教育和培训

学习资源通用包装

1 范围

本标准规定了一种学习资源的通用包装格式，提出了学习资源通用包装包的概念模型、信息模型、组成要素及基本要求。

本标准适用于学习资源的描述、包装和发布。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB 13000-2010 信息技术 通用多八位编码字符集（UCS）

RFC 3986 统一资源标识符（URI）：通用句法（Uniform Resource Identifier (URI): Generic Syntax）

3 术语和定义

下列术语和定义适用于本文件。

3.1

通用包装 common cartridge

一种为学习资源的标记、封装、分发和访问提供统一格式的学习资源内容包装标准。

注：为描述简便，以下本标准名称简称为CC。

3.2

通用包装包 common cartridge package

采用本标准制作和封装的内容包。

3.3

从属元素 dependency element

标记某个特定类型的资源以及资源类型之间某些关系的元素。

3.4

目录 directory

内容包中文档的物理文件夹。

3.5

交换包 interchange package

系统间交换的组件集合。

注：本标准使用“交换包”时，它总是包括内容清单，并且可以包括内容文件和控制文件。所有包含在交换包里的文件都在内容清单中描述。对于逻辑包而言，当逻辑包里的所有组件都位于该交换包本地时，交换包就等同于该逻辑包。

3.6

可运行的URI launchable URI

可运行的URI是对包含在资源描述中并且可用于定位和访问由资源描述的内容的统一资源定位符（URL）的表示。

注：可运行的URI不能被包阅读器解析。在处理完交换包内容之后，URI不能被存储，以供其他软件组件（例如，Web浏览器）使用。

3.7

学习模块 learning module

学习内容和/或应用程序功能的集合，用于表述学习活动或是学习活动的一部分。

3.8

学习资源 Learning Resource

用于学习、教育或培训的数字或非数字的任何实体。

3.9

本地组件 local componet

交换包包含的逻辑包的任何组件。

3.10

素材 material

所有提供给用户的而不是由系统处理的静态文本、图象或媒体对象。

3.11

命名空间 namespace

一组名称，这些名称中每个都是唯一的且有不同于同组中其他名称的语义值。

3.12

包 package

一个逻辑目录，包含一个特定名称的内容清单文件以及内容清单文件中引用的实际资源文件。

3.13

包交换文件 package interchange file

由包组合成的一个可供交换的文件，是一种通过网络传输的结构化信息的简便形式。

3.14

包阅读器 package reader

一种读取内容清单和验证交换包内容的软件组件。

注：包阅读器可以处理逻辑包（如，检索和存储内容清单引用的信息、拆封来自包交换文件的本地文件、检索或登录远地文件的地址等）或代表另一个软件组件执行任务。

3.15

包写入器 package writer

一种软件组件，它创建或修改交换包实例、装配内容文件和本地向交换包声明的其他文件并将它们写入绑定的目标交换包，或代表其他软件组件执行此类任务。

3.16

题项 question item

一套试题中的单条题目，是最小可交换的评估对象。

3.17

试题测试互操作 Question Test Interoperability

一种用于表征问题和测试数据及其相应结果报告的数据交换格式。

3.18

相对引用 relative reference

一种通过描述对引用上下文的扩展来引用资源的URL。

注：扩展和上下文组合构成目标URI。

3.19

远地组件 remote componet

处于交换包以外的逻辑包的任何组件。

3.20

目标系统 target system

导入通用包装包的学习管理系统或类似系统。

4 缩略语

CC: 通用包装 (Common Cartridge)
CELTS: 中国网络教育技术标准 (Chinese E-Learning Technology Standards)
DHTML: 动态超文本置标语言 (Dynamic Hyper Text Mark-up Language)
GUID: 全局唯一标识符 (Globally Unique Identifier)
HTML: 超文本置标语言 (Hyper Text Mark-up Language)
HTTP: 超文本传输协议 (Hyper Text Transfer Protocol)
IP: 互联网协议 (Internet Protocol)
LAO: 学习应用对象 (Learning Application Object)
LMS: 学习管理系统 (Learning Management System)
LOM: 学习对象元数据 (Learning Object Metadata)
LTI: 学习工具互操作 (Learning Tools Interoperability)
LUID: 本地唯一标识符 (Locally Unique Identifier)
PIF: 包交换文件 (Package Interchange File)
UML: 统一建模语言 (Unified Modeling Language)
URI: 统一资源标识符 (Uniform Resource Identifier)
URL: 统一资源定位符 (Uniform Resource Locator)
Web: 万维网 (World Wide Web)
XML: 可扩展置标语言 (Extensible Markup Language)
XSD: XML结构定义 (XML Schemas Definition)

5 通用包装概念模型

5.1 通用包装概念图

通用包装包的一些特征与现有商业解决方案中的“类包装包”相似。从概念上说，通用包装包是可整合到LMS学习环境的文件包及其元数据。在高阶层次上，可直接和LMS中的“课程”这一概念相对应。通用包装的概念模型见图1。



图1 通用包装概念模型图

图1所示的概念模型图中，逻辑包是一个或多个可用（或可重用）内容单元的表现，包含内容清单描述的全部组件，包括本地的和通过引用纳入的外部组件。

逻辑包包含交换包和外部内容两大部分：

- 交换包：本标准规定的学习内容包装的主体是交换包，提供者通过交换包为各类人群提供适当的学习内容，使用者通过交换包交换相关信息。交换包是在系统之间交换的组件集合，包括内容清单、学习应用对象、包级 Web 内容和具体计算机文件。其实例是包交换文件，包交换文件是一个压缩文件，其实现可采用，例如，zip、rar 和 jar 等常见的文件压缩格式，本标准不规定具体采用何种压缩格式；
- 外部内容：概念模型中定义了若干外部组件，表明本标准定义的概念模型将支持可用于定义新的词汇和结构的扩展机制。

5.2 概念模型描述

5.2.1 交换包结构

通用包装的交换包结构见图2。

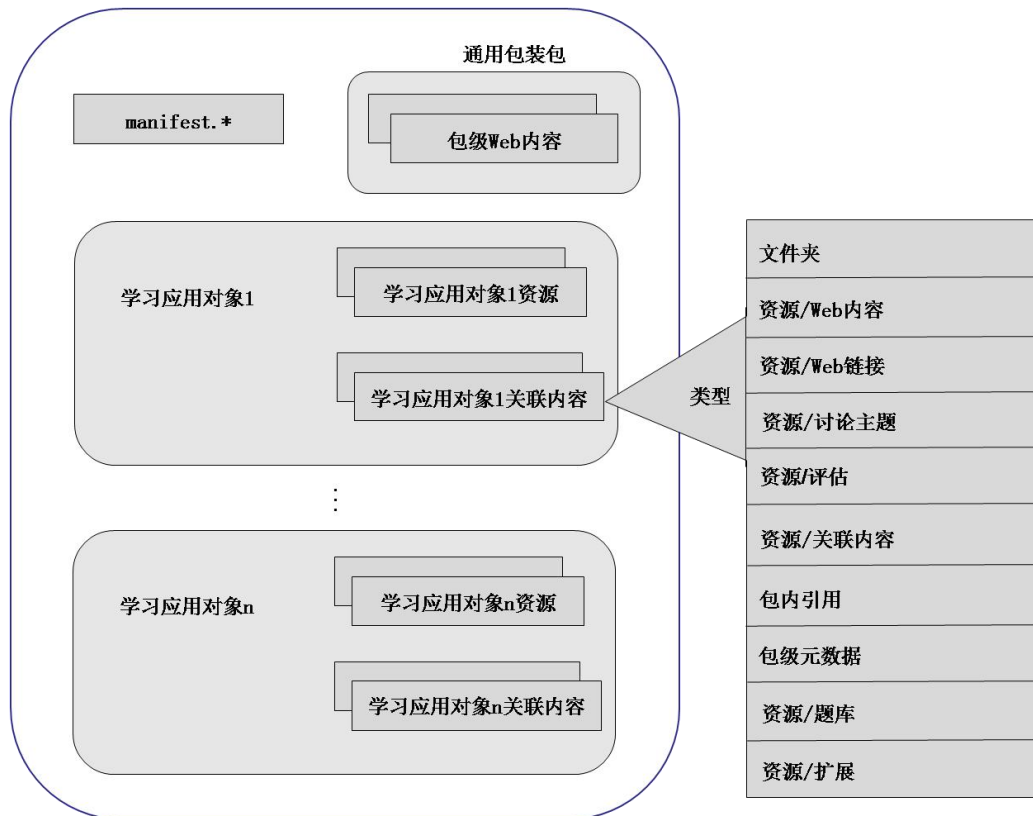


图 2 交换包结构

包交换文件由内容清单、n个学习应用对象和包级Web内容等大类内容构成：

- a) 内容清单：manifest 是一个内容清单文件。对于 manifest.*，如果用 XML 实现，则为 manifest.xml。按本标准，所有内容都在根目录下的内容清单文件中定义，不使用子内容清单，子内容清单是指从属于该清单的另一个完整清单；
- b) 学习应用对象：是任何一种资源类型，在被导入到目标系统并进一步表征之前，需要额外的处理和解释。物理上，一个学习应用对象由内容包中该对象专属的一个目录和子目录组成。每个学习应用对象文件夹包括学习应用对象资源（以描述符文件呈现）及可选的关联内容：
 - 1) 描述符文件：作为入口点的文件，用于访问一个学习应用对象的信息，以便将学习应用对象导入到目标系统中。描述符文件通常为一个 XML 文件，该文件符合基于学习应用对象类型的特定的文件规范。但在某些情况下，文件也可以是一个 zip 文档或者其他结构化格式的文档。描述符文件不是要在目标系统内呈现，而是要在导入通用安装包时被目标系统处理；
 - 2) 关联内容：是一种资源类型，包含某个特定学习应用对象的文件集合。每个被引用的文件应存在于包含与其相关的学习应用对象的描述符文件所在的目录下，或其任何一个子目录下。关联内容包括的内容类型有：文件夹、Web 内容资源、Web 链接资源、讨论主题资源、评估资源、关联内容资源、包内引用、通用安装包包级元数据、题库资源和扩展资源等。
- c) 包级 Web 内容：一个包级 Web 内容文件夹可包含在一个通用安装包内。若包级 Web 内容文件夹存在，则应位于包的根目录下，本标准不对该文件夹的命名进行定义。在包级 Web 内容文件夹中的 Web 内容可被包内任何一个学习应用对象引用。允许该文件夹及其子文件夹内的文件采用相对路径引用，但不允许该文件夹内的文件引用一个学习应用对象文件夹或其关联内容文件夹内的文件。

5.2.2 内容类型

5.2.2.1 文件夹

文件夹是以特定顺序存放内容项和子文件夹的简单集合；其中的子文件夹是嵌套的文件夹，即包含的其他文件夹。基于内容项的文件夹代表一种内容呈现方式，可用于定义如何组织内容并呈现给学习者。

5.2.2.2 资源/Web 内容

是一种资源类型。Web内容可以是所有能在网络上传播的文件，包括HTML文件、图片、音频、视频等。HTML文件中可包含指向其他Web内容文件的链接，这些文件既可在包内也可在包外。

5.2.2.3 资源/Web 链接

是一种资源类型。Web链接以一个标准HTTP链接形式表达学习应用对象，通过添加标题来扩展标准的HTTP链接。它还包含一些属性用来描述资源在哪个窗口打开，以及打开的其他窗口的特征，如窗口尺寸。

5.2.2.4 资源/讨论主题

是一种资源类型，用于启动讨论活动的一种学习应用对象。它表述一个讨论主题的占位符，存放用于生成讨论主题的初始信息，但并非指向外部系统现有讨论主题的链接。在导入LMS后，希望仅利用内部工具来生成一个新的讨论主题。它的属性包含标题、描述、附件等。

5.2.2.5 资源/评估

是一种资源类型。评估类资源是QTI评估实例，可包含通用包装支持的任何问题类型。评估包含的属性包括，如尝试次数、时间限制及是否允许延期提交等。

5.2.2.6 资源/关联内容

资源/关联内容是学习应用对象文件夹中的“关联内容”下面的特定学习应用对象的关联内容资源，也是一种资源类型，此类资源是供单个学习应用对象专用的文件集合。

关联内容资源包含特定学习应用对象使用的文件集。所引用的每个文件都存在于包含与之关联的学习应用对象的描述符文件所在目录或其子目录下。此外，每个关联内容资源与一个且只与一个特定学习应用对象关联。

5.2.2.7 包内引用

包内引用是实现关联的一种方式，允许学习应用对象或包文件引用本包内的其他文件。

5.2.2.8 包级元数据

是通用包装包级别的元数据，包括授权、接口、描述等元素。在manifest.*文件中列出。

5.2.2.9 资源/题库

是一种资源类型，其表现形式是QTI对象库实例。每个通用包装包中最多包含一个题库。可以包含通用包装支持的所有QTI题型。题库中的题项不供通用包装包中任何评估引用。

通用包装至少支持以下题型：

- 选择题（单选）；
- 选择题（多选）；

- 是非题；
- 填空题（答案框只有一个正确答案）；
- 匹配题（提供单一答案框，但也支持通用包装包中“容器”类组件和正则表达式的匹配）；
- 论述题。

题项可以只包含在通用包装包中，也可以作为评估资源或题库资源的一部分。

通常，一条题项包含以下内容：

- 题项的标签/标题；
- 题项的文本（可包含 HTML、包内引用、URL、样式等）；
- 答案选项（可包含 HTML、包内引用、URL、样式、图片、视频音频等）；
- 题项答案得分；
- 反馈（可包含 HTML、包内引用、URL、样式、图片、视频音频等）；
- 答案显示设定；
- 答案提示（可包含 HTML、包内引用、多个 URL、样式、图片、视频音频等）；
- 题项设置参数（如时间等）；
- 题项元数据。

5.2.2.10 资源/扩展

扩展类资源是提供补充功能的且不在本通用包装范围内但在 CELTS 注册的资源。

5.2.3 支持的资源类型

通用包装包支持的资源类型见表1。

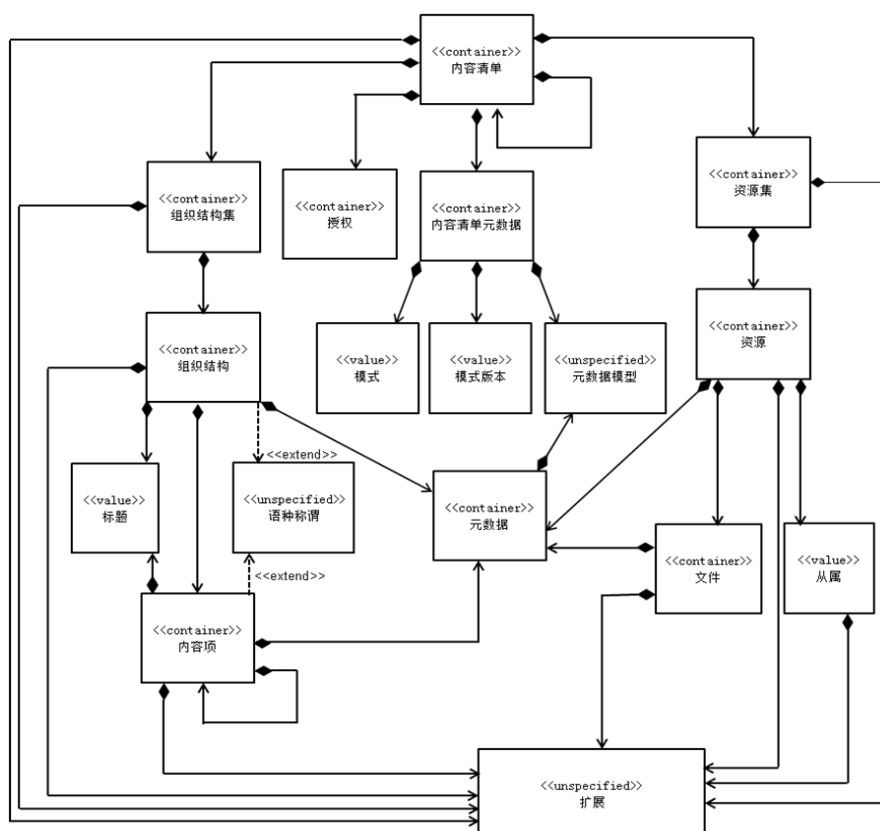
表1 通用包装包支持的资源类型

资源类型	限制
Web 内容	0 个或多个
关联内容	0 个或多个
评估（通用包装）	0 个或多个
题库（通用包装）	0 个或1个
授权数据	0 个或1个
讨论主题	0 个或多个
Web 链接	0 个或多个

6 通用包装信息模型

6.1 通用包装信息模型结构

通用包装信息模型结构见图3。



注：模型组件在图中以方框表示，组件间关系用带箭头的线表示（实心菱形表示起点，虚线表示可能的扩展关系）。箭头的起点和终点与方框的连接点是针对整个大框的。图中双尖括号（《》）中的英文词表示组件类型，组件类型下面的文字是归类于该类组件的一个实例。

图3 通用包装信息模型结构

通用包装信息模型的组件分为容器、值、未指明和特性四大类型：

- 容器（container）：一个容器类组件可能包含一个或多个其他容器类组件或信息模型其他大类的组件。例如，内容清单这个实例组件中承载的信息支持其他各类容器中的组件，也支持该大类中的组件。在相关的描述中，这种支持和被支持的关系称为父辈和子辈；
- 值（value）：这个大类的组件没有子辈组件，即，它不应由容器、值、未指明或特性等大类中的组件合成。值类组件是容器类组件的子辈，并且其语义值在其父辈类组件的语义值范围内；
- 未指明（unspecified）：当前尚不确切的组件（备用类）或被认为不重要的组件。未指明类组件可以是父辈的。未指明类充当信息模型的扩展点；
- 特性（characteristic）：图中没有明确示出特性类及其组件实例。特性类组件分布于各个容器中。此类组件的信息是容器类组件的修饰类信息，它没有下属组件。特性类组件用于声明容器类组件的品质或值，是容器类组件的内在特征或容器类构建特征的一部分。特性类组件与它修饰或实际描述的容器类组件紧密耦合。

6.2 信息模型组件

6.2.1 组件描述格式

通用包装信息模型中容器、值、未指明和特性类等大类范畴的各组件采用表2描述格式，即通过该组件的名称开头的若干描述符及相应描述符的定义或说明予以描述。

表 2 通用包装信息模型组件描述格式

描述符	定义/说明
组件名称	所描述的组件的英文名称。
组件类型	该组件所属的大类名称（容器、值、未指明或特性中的一个）。
数据类型	<p>用于值和特性类，是所描述组件的有效值的允许结构。</p> <p>有效的数据类型如下：</p> <ol style="list-style-type: none"> 1. URI：符合 RFC 3986 定义的 URI 句法的有效实例。 2. LUID：内容清单中本地唯一标识符。它以附带值空间约束的字符串数据类型为基础。 3. LUIDref：内容清单里任何地方定义的对 LUID 的引用。LUID 和引用该 LUID 的 LUIDref 的值相同。 4. 布尔型：基本二值数据类型，使用“true”（真）和“false”（假）这两个词指出对象的逻辑状态。 5. 字符串：字符序列。 6. 未指明：不知道的或不重要的数据类型。
值空间	所描述组件的有效值的范围。如果值空间未指明，就表示这个值空间不知道或不重要。
多重性	<p>所描述组件的一种性质，它指出可能在给定父辈环境下使用或出现的次数。此性质的值使用下面的范围标注或范围代号表达（括号里是对值范围标注的解释）：</p> <p>‘0..1’ [任选；受限] ‘0..未定界’ [任选；不受限] ‘1..1’ [强制；受限] ‘1..未定界’ [强制；不受限]</p> <p>在 UML 模型中，多重性也可以用代号标注方式表达。相应的代号应是（括号里的解释除外）：</p> <p>‘*’ [任选；不受限] ‘1’ [强制；受限] ‘1..*’ [强制；不受限]</p> <p>如果多重性大于 1，则通过附加“，” <i>有序的</i>或“，” <i>无序的</i>来指出同辈排序的重要性。 <i>有序的</i>的用列表规定同辈序列。<i>无序的</i>的规定一个同辈的集合或口袋，顺序本身不重要。</p>
特性类	<p>使用“{” characteristic * “，” characteristic “}”的形式列出与所描述组件关联的各特性类组件。可以在花括号内表达一个或多个特性类组件。各个特性类组件之间应使用逗号（，）隔开。</p> <p>如果列出的特性类组件不止一个，则通过附加“，” <i>有序的</i>或“，” <i>无序的</i>来指出同辈排序的重要性。 <i>有序的</i>的用列表规定同辈序列。<i>无序的</i>的规定一个同辈的集合或口袋，顺序本身不重要。</p> <p>本标准规定的特性类组件包括：基础(Base)、默认(Default)、超文本引用(Href)、标识符(Identifier)、标识符引用(IdentifierRef)、可视性(IsVisible)、语言(Language)、其他(Other)、参数(Parameters)、结构(Structure)、类型(Type)、版本(Version)。</p>
父辈	列出可能是该组件的父辈的各组件。
子辈	<p>使用“[” child * “，” child “]”的形式列出该组件的可能的各个子辈组件。可以在方括号内表达一个或多个子辈。各个子辈之间应使用逗号（，）隔开。</p> <p>如果列出的子辈不止一个，则通过附加“，” <i>有序的</i>或“，” <i>无序的</i>来指出同辈排序的重要性。 <i>有序的</i>的用列表规定同辈序列。<i>无序的</i>的规定一个同辈的集合或口袋，顺序本身不重要。</p>
描述	包含所描述的组件和该组件的值空间有关的各个描述。
<p>注：如果所描述的组件存在子辈组件，其定义用方括号中加相应的英文名称表示。如，InterchangePackage有子辈类Manifest，相应的定义/说明栏中表示为[Manifest]；如果某个描述符的“定义/说明”不适用，用“--”表示；在描述中出现的英文单词、连词和/或组合词，每个单词都以大写字母开头，它们代表信息模型中某个抽象组件。</p>	

6.2.2 信息模型组件描述

6.2.2.1 信息模型组件名称及类型

通用包装信息模型的组件名称及类型见表3。各类组件的定义在后续各条中给出。

表3 通用包装信息模型组件

编号	组件名称	英文名称	所属类型
1	交换包	InterchangePackage	容器
2	内容清单	Manifest	容器
3	内容清单元数据	ManifestMetadata	容器
4	模式	Schema	值
5	模式版本	SchemaVersion	值
6	元数据模型	MetadataModel	未指明
7	组织结构集	Organizations	容器
8	组织结构	Organization	容器
9	标题	Title	值
10	语种称谓	LingualTitle	未指明
11	内容项	Item	容器
12	资源集	Resources	容器
13	资源	Resource	容器
14	文件	File	容器
15	从属	Dependency	值
16	授权	Authorizations	容器
17	元数据	Metadata	容器
18	扩展	Extension	未指明
19	基础	Base	特性
20	默认	Default	特性
21	超文本引用	Href	特性
22	标识符	Identifier	特性
23	标识符引用	IdentifierRef	特性
24	可视性	IsVisible	特性
25	语言	Language	特性
26	其他	Other	特性
27	参数	Parameters	特性
28	结构	Structure	特性
29	类型	Type	特性
30	版本	Version	特性

注：对于本标准，上述各类组件涉及到的元数据均为本标准规定的学习对象元数据（简称LOM）。

6.2.2.2 交换包

交换包（InterchangePackage）的定义见表4。

表4 InterchangePackage 定义

描述符	定义/说明
组件名称	InterchangePackage
组件类型	容器
数据类型	--
值空间	--
多重性	1..1
特性类	--
父辈	无, InterchangePackage是最高层组件, 没有父辈组件
子辈	[Manifest]
描述	<p>InterchangePackage 对象是系统之间交换的逻辑包的子集。InterchangePackage 应满足下列条件:</p> <ol style="list-style-type: none"> 1. InterchangePackage 应包括一个 Manifest 对象并且可以包含内容文件和控制文件。 2. Manifest 中使用 URI (解析到 InterchangePackage 里某个位置) 描述的任何文件应包含在该 InterchangePackage 中。 3. InterchangePackage 中包含的所有文件应在该 Manifest 中描述。 4. InterchangePackage 中包含的文件可包含对其他文件的内部引用(如,HTML 内容文件可以引用 JPEG 内容文件,或者,XML Schema 控制文件可以引用另一个 XML Schema 控制文件)。如果 InterchangePackage 中包含的文件使用 URI (解析到 InterchangePackage 里某个位置) 引用另一个文件, 所引用的文件应在 InterchangePackage 的 Manifest 中描述。 <p>PIF 是 InterchangePackage 的一个具体实例。包阅读器和包写入器不负责支持包交换文件的读出或写入。PIF 应满足下列条件:</p> <ol style="list-style-type: none"> 1. PIF 应在其根上包含一个内容清单对象, 该对象应限定为 XML 文档。 2. PIF 中包含的任何控制文件应放置在该 PIF 的根上。 3. 从根内容清单文档到包含在 PIF 中包含的文件的所有引用都应通过相对引用执行。这些引用应是相对于该 PIF 的根的。 4. PIF 不应包含处于以下情况的任何文件: 绝对路径(由相对引用声明或解析得到的)高于同一个路径层次中内容清单对象的路径, 或者, 文件的绝对路径完全不同于内容清单的位置。

6.2.2.3 内容清单

内容清单 (Manifest) 的定义见表5。

表5 Manifest 定义

描述符	定义/说明
组件名称	Manifest
组件类型	容器
数据类型	--
值空间	--
多重性	1..1 作为InterchangePackage的子辈 0..不受限 作为Manifest的子辈, 无序
特性类	{Identifier, Version, Base, Other}, 无序

表 5 （续）

描述符	定义/说明
父辈	InterchangePackage
子辈	[ManifestMetadata, Organizations, Resources, Extension], 有序
描述	<p>Manifest对象是描述逻辑包完整实例的数据结构的容器。</p> <p>Manifest可以包含对本地或远地组件的引用，这里的本地和远地是针对组件的父辈InterchangePackage对象而言。可通过Resource、File对象引用。Resource和File用于引用内容文件和控制文件。</p> <p>Manifest应包含描述所有为解释该Manifest所需的控制文件的File对象。</p>

6.2.2.4 内容清单元数据

内容清单元数据（ManifestMetadata）的定义见表6。

表 6 ManifestMetadata 定义

描述符	定义/说明
组件名称	ManifestMetadata
组件类型	容器
数据类型	--
值空间	--
多重性	0..1
特性类	--
父辈	Manifest
子辈	[Schema, SchemaVersion, MetadataModel], 有序
描述	<p>ManifestMetadata对象包含对其父辈Manifest对象的描述信息。ManifestMetadata涉及的范围是其父辈Manifest描述的整个逻辑包。</p> <p>ManifestMetadata的Schema和SchemaVersion子辈提供规范或配置文件相关信息来管理父辈Manifest中的含义。ManifestMetadata中的MetadataModel子辈作为一个占位符用于对其父辈Manifest的一般性描述信息。MetadataModel是一个扩展点，允许元数据具有在其他命名空间定义的信息结构。多个不同的元数据模型可作为单个ManifestMetadata对象中的扩展被声明。</p> <p>允许MetadataModel和外部引用元数据的任一组合。</p> <p>ManifestMetadata有通用包装包级元数据和角色元数据两种，包级元数据（位于根目录下）是强制要求的，角色元数据是可选的。ManifestMetadata可以在内容清单中用于定义哪一类用户可以访问特定的资源。</p> <p>1. 包级元数据</p> <p>包级元数据元素以及它的模式和模式版本应在内容清单中定义。</p> <p>任何媒体播放器、编解码器、浏览器插件或者运行系统对通用包装包的内容的要求都应在包级元数据的描述中声明。每个条目都应该包括工具、产品名称、版本号、供应商名称和URL等详细信息。这种要求应以自由文本的形式记入到包的描述元素中。</p>

表 6 （续）

描述符	定义/说明
描述	<p>2. 角色元数据</p> <p>在某些情况下，资源需要在组织结构内指定，而不应在默认导入包的时候在播放器模式下可见，例如，教师手册、教案、教学笔记及解决方案文档等只有教师可见。另一种情况是出版商可能提供额外的可选资源，这些资源可以由教师在以后有选择地向学生发布。无论哪一种情况，即使初始时在播放通用包装包时资源对于学习者来说是不可见的，都需要在组织结构内指出资源应在哪里出现。这些资源应在编辑通用包装包时可见，以便在适当时可以修改设置。</p> <p>为满足以上需求，通用包装包在内容清单文件中提供了可选的角色（包括学生、教师、教学设计师、管理员、LMS）元数据与文件中的资源的关联。如不特别声明，则资源默认向所有用户开放。如果显式声明，则可以指定资源对哪些角色可见。</p> <p>注意：指定一个角色会对一个学习平台行为产生影响。例如，针对教师的内容不应对学生可见。</p>

6.2.2.5 模式

模式（Schema）的定义见表7。

表 7 Schema 定义

描述符	定义/说明
组件名称	Schema
组件类型	值
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0..1
特性类	--
父辈	ManifestMetadata
子辈	--
描述	<p>Schema 对象声明其父辈 Manifest 对象的规范或配置文件的名称。Schema 类的值告知包阅读器了解规范或配置文件的方式，从而管理 Manifest 文件的含义。该值不应与分配给一个给定元数据模式的名称或标签混淆。</p> <p>对于 Schema 的值，除了不应包含任何版本信息之外，对于字符串的内容或语法没有要求。</p>

6.2.2.6 模式版本

模式版本（SchemaVersion）的定义见表8。

表 8 SchemaVersion 定义

描述符	定义/说明
组件名称	SchemaVersion
组件类型	值
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0..1

表 8 （续）

描述符	定义/说明
特性类	--
父辈	ManifestMetadata
子辈	--
描述	<p>SchemaVersion 对象声明规范或配置文件的版本，SchemaVersion 的值用于声明其同辈 Schema 对象。该值将一个同辈 Schema 标识的模式或配置文件的版本告知包阅读器。一个 SchemaVersion 的值不应包含任何其他信息。</p> <p>本标准既不指定版本信息的语法也不指定应用任何版本语法的启发式规则。</p>

6.2.2.7 元数据模型

元数据模型（MetadataModel）的定义见表9。

表 9 MetadataModel 定义

描述符	定义/说明
组件名称	MetadataModel
组件类型	未指明
数据类型	未指明
值空间	未指明
多重性	0.. 不受限，排序也未指明
特性类	未指明，排序也未指明
父辈	ManifestMetadata Metadata
子辈	未指明，排序也未指明
描述	<p>MetadataModel 对象是一个占位符，将交换包中包含的元数据的有效位置告知这个信息模型的绑定（语言）。</p> <p>包含一个或多个元数据模型的扩展容器或值组件类型的真正名称，只有在组件的绑定被导入到信息模型的一个实例中时才会知道，因此组件的实际类型在本信息模型中未指明。</p> <p>MetadataModel 由两个元数据容器使用：ManifestMetadata 和 Metadata。在本标准的一个绑定实例中，MetadataModel 是表达元数据模型及其相关信息的唯一方法。</p> <p>一个包写入器可表述为不同元数据模型，这些元数据模型的种类应尽足以用于描述由 MetadataModel 的父辈对象封装的内容所需的元数据模型。</p> <p>MetadataModel 仅用于声明一个或多个元数据模型及其相关信息。MetadataModel 的语义应遵守该信息模型中它的父辈语义。MetadataModel 的语义不应覆盖或重定义该信息模型中定义的其父辈的语义。</p>

6.2.2.8 组织结构集

组织结构集（Organizations）的定义见表10。

表 10 Organizations 定义

描述符	定义/说明
组件名称	Organizations
组件类型	容器
数据类型	--
值空间	--
多重性	1..1
特性类	{Default, Other}, 无序
父辈	Manifest
子辈	[Organization, Extension], 有序
描述	Organizations 对象是描述 Resources 对象之间逻辑关系的类的容器, 是多个 Organization 的集合。

6.2.2.9 组织结构

组织结构 (Organization) 的定义见表11。

表 11 Organization 定义

描述符	定义/说明
组件名称	Organization
组件类型	容器
数据类型	--
值空间	--
多重性	0..不受限, 有序
特性类	{Identifier, Structure, Other}, 无序
父辈	Organizations
子辈	[Title, LinguageTitle, Item, Metadata, Extension], 有序
描述	Organization 对象是类的容器, 这些类用于描述由 Manifest 对象封装的 Resource 对象之间特定逻辑关系。各 Organization 对象都有相同的目标。对于给定 Manifest 中的同一 Resource 对象集合, 每个对象显示一种构造方法, 这样每个 Organization 会显示一个不同的结构。 Item 子辈对象用于表达 Organization 里的结构节点。

6.2.2.10 标题

标题 (Title) 的定义见表12。

表 12 Title 定义

描述符	定义/说明
组件名称	Title
组件类型	值
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0..1
特性类	--

表 12 (续)

描述符	定义/说明
父辈	Organization Item
子辈	--
描述	Title 对象包含一个文本值应用于 Organization 或 Item 对象来命名或标记它们所表达的结构。文本值没有语言类型。Title 的取值是一个字符串。如果希望在标题中实现一个或多个语言敏感的字符串表达, 则应采用 LingualTitle 类。

6.2.2.11 语种称谓

语种称谓 (LingualTitle) 的定义见表13。

表 13 LingualTitle 定义

描述符	定义/说明
组件名称	LingualTitle
组件类型	未指明
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0.. 不受限, 无序
特性类	{Language}
父辈	Organization Item
子辈	--
描述	LingualTitle 对象包含一个应用于 Organization 或 Item 对象的特定语言的文本值用以命名或标记它们所表达的结构。LingualTitle 类的语言由 Language 对象制定。

6.2.2.12 内容项

内容项 (ContentItem) 的定义见表14。

表 14 ContentItem 定义

描述符	定义/说明
组件名称	ContentItem
组件类型	容器
数据类型	--
值空间	--
多重性	1.. 不受限, 有序
特性类	{Identifier, IdentifierRef, IsVisible, Parameters, Other}, 无序
父辈	Organization Item
子辈	[Title, Item, Extension], 有序

表 14 (续)

描述符	定义/说明
描述	<p>ContentItem 对象是一个容器, 用于表达一个特定 Organization 内的结构节点或另一个 ContentItem 对象中的结构节点。</p> <p>ContentItem 可使用一个 IdentifierRef 对象来表达对一个 Resource 对象(与其结构化的角色和位置有关)的内部引用。</p> <p>在本标准中内容项英文简称为 Item。</p>

6.2.2.13 资源集

资源集 (Resources) 的定义见表15。

表 15 Resources 定义

描述符	定义/说明
组件名称	Resources
组件类型	容器
数据类型	--
值空间	--
多重性	1..1
特性类	{Base, Other}, 无序
父辈	Manifest
子辈	[Resource, Extension], 有序
描述	<p>Resources 对象是一种 Manifest 使用的内容和文件的详细目录, 是多个 Resource 的集合。</p> <p>Resources 对象是关于父辈 Manifest 对象使用的文件的所有信息的容器。这些文件可以是本地的或远地的。</p> <p>Resources 描述的文件是用于其父辈 Manifest 的。任何其他 Manifest 对象(无论它们是同辈、子辈或父辈 Manifest 的先辈)中的引用文件, 都不在 Resources 对象范围内。</p>

6.2.2.14 资源

资源 (Resource) 的定义见表16。

表 16 Resource 定义

描述符	定义/说明
组件名称	Resource
组件类型	容器
数据类型	--
值空间	--
多重性	0..不受限, 无序
特性类	{Type, Base, Href, Other}, 无序
父辈	Resources
子辈	[Metadata, File, Dependency, Extension], 有序

表 16 (续)

描述符	定义/说明
描述	Resource 对象是一个信息容器，所述信息与先辈 Manifest 对象使用的文件的特定汇集有关。 Resource {Href} 对象声明的值是一个可运行的URI。 Resource {Href} 中声明的文件引用应与同一个Resource里某File对象中的声明相关联。 所关联的File对象的Href应引用与Resource {Href} 所引用的相同的文件。 Resource {Href} 与所关联的File {Href} 的URI可能不同，因为Resource {Href} 可能在URI中包含“运行”参数。

6.2.2.15 文件

文件 (File) 的定义见表17。

表 17 File 定义

描述符	定义/说明
组件名称	File
组件类型	容器
数据类型	--
值空间	--
多重性	0.. 不受限，无序
特性类	{Href, Other}, 无序
父辈	Resource
子辈	[Metadata, Extension], 有序
描述	File对象是与单个计算机文件相关的所有信息的容器，由其父辈Resource对象封装。File保存描述计算机文件的元数据和对文件位置的引用。

6.2.2.16 从属

从属 (Dependency) 的定义见表18。

表 18 Dependency 类定义

描述符	定义/说明
组件名称	Dependency
组件类型	值
数据类型	--
值空间	--
多重性	0.. 不受限，无序
特性类	{IdentifierRef, Other}, 无序
父辈	Resource
子辈	[Extension]

表 18 (续)

描述符	定义/说明
描述	<p>Dependency对象允许一个Resource对象引用同辈Resource对象中描述的文件集。</p> <p>Dependency应使用IdentifierRef对象来引用Resource。引用的Resource应由Dependency对象的祖父辈Resources对象封装。</p> <p>引用的结果是，在被引Resource范围内的文件，应考虑进引用的Dependency的父辈Resource的范围内。与引用Resource相关联的特征对象不应被考虑进引用的Dependency父辈Resource的范围内。</p>

6.2.2.17 授权

授权 (Authorizations) 的定义见表19。

表 19 Authorizations 定义

描述符	定义/说明
组件名称	Authorizations
组件类型	容器
数据类型	--
值空间	--
多重性	1..1
特性类	--
父辈	无
子辈	无
描述	<p>通用包装规定的 Authorizations 支持以下概念：</p> <ol style="list-style-type: none"> 需要导入通用包装包的 Authorizations。 需要使用通用包装包的 Authorizations。 需要使用通用包装包内特定资源的 Authorizations。 <p>其中 2 和 3 是互斥的，也就是说，一个通用包装包要么指定包内所有资源被保护要么指定部分资源被保护。</p> <p>本标准对授权访问特定资源的机制不进行定义。如何将权限征询的机制整合到用户学习过程中由 LMS 的具体实现决定。例如，当用户访问包含受限资源的某一课程时，LMS 需要征询用户的权限。或者，LMS 在用户试图访问课程受限资源时进行征询。</p>

6.2.2.18 元数据

元数据 (Metadata) 的定义见表20。

表 20 Metadata 定义

描述符	定义/说明
组件名称	Metadata
组件类型	容器
数据类型	--
值空间	--
多重性	0..1

表 20 (续)

描述符	定义/说明
特性类	--
父辈	Organization Item Resource File
子辈	[MetadataModel]
描述	Metadata 对象包含其父辈容器类型对象的描述信息。Metadata 的范围仅涵盖父辈容器类。 MetadataModel 子辈对象作为一个容器来描述其父辈的信息。MetadataModel 是一个扩展点，允许元数据具有其他命名空间定义的信息结构。多个不同的元数据模型可以声明为扩展，所述扩展包含在单个 MetadataModel 对象内。

6.2.2.19 扩展

扩展 (Extension) 的定义见表21。

表 21 Extension 定义

描述符	定义/说明
组件名称	Extension
组件类型	未指明
数据类型	未指明
值空间	未指明
多重性	0.. 不受限，有序
特性类	未指明，排序也未指明
父辈	Manifest Organizations Organization Item Resources Resource File Dependency
子辈	未指明，排序也未指明
描述	<p>Extension 对象是一个占位符。它将值类或容器类组件的有效位置告知信息模型的绑定从而扩展信息模型中的任何容器类组件。</p> <p>Extension 组件是信息模型中扩展容器类的任何组件的两种机制之一。第二种扩展机制是 Other 类。</p> <p>Extension 对象用于扩展容器类的组件。用于替换一个 Extension 对象的扩展容器类或值类组件的实际名称，只有在该对象的绑定被导入到信息模型的实例时才会知道。因此，此信息模型中组件的实际类型未指明。</p> <p>Extension 的语义应在信息模型中其父辈对象的语义范围内。Extension 的语义不应覆盖或重新定义信息模型中已经定义了的任何父辈对象的语义。</p>

6.2.2.20 基础

基础（Base）的定义见表22。

表 22 Base 定义

描述符	定义/说明
组件名称	Base
组件类型	特性
数据类型	URI
值空间	与绑定语言相关
多重性	0..1
父辈	Manifest Resources Resource
描述	<p>Base 对象用于指定与其关联的对象的基准 URI，以便解析在关联对象后辈中的相对引用。</p> <p>Base 组件的取值范围应适用于与其关联对象的所有后辈对象，除非 Base 的值被经允许的后辈的后续 Base 对象替换或修改。</p> <p>对 Base 来说，所有相对路径段应解析到包含这些路径的内容清单文档。相对路径段将通过以下方式构造：Base 表述的路径段能够附加到距离其最近的祖辈中的 Base 对象所表达的相对路径段上，直至根 Manifest 对象。之后这一系列相对路径段应被附加到内容清单文件所在位置或被处理生成绝对路径。</p>

6.2.2.21 默认

默认（Default）的定义见表23。

表 23 Default 定义

描述符	定义/说明
组件名称	Default
组件类型	特性
数据类型	与绑定语言相关（默认是字符串）
值空间	组织结构集的值 {Default}。组织结构 {标识符} 对象
多重性	0..1
父辈	Organizations
描述	<p>Default 对象指定一个 Organizations 对象中的子辈 Organization 对象作为给定 Manifest 对象的主要或默认的组织结构。</p> <p>应通过引用目标 Organization 的 Identifier 对象的值来指定默认 Organization。目标 Organization 应是具有 Default 的 Organizations 对象的子辈。不允许 Default 的其他引用。</p> <p>如果一个 Organizations 对象未声明 Default 值时，则 Organizations 里最先定义的 Organization 对象将被视为主要或默认的组织结构。</p>

6.2.2.22 超文本引用

超文本引用（Href）的定义见表24。

表 24 Href 定义

描述符	定义/说明
组件名称	Href
组件类型	特性
数据类型	URI
值空间	根据 RFC3986 定义
多重性	0..1: 作为Resource的子辈 1..1: 作为File的子辈
父辈	Resource File
描述	<p>Href 对象用于定位资源。</p> <p>Href 声明的值语法上应为有效的 URI。本标准不保证所声明的字符串是有效的 URI，或被解析为所声明位置的实际资源或资源的一部分。</p> <p>Manifest.Resources.Resource {Href} 声明的值表述为一个 URL，可用于定位和访问该资源所描述的内容。包阅读器不需要解析该 URI。该 URI 是要被存储下来以便稍后在一个交换包内容被处理后由其他软件部分来使用。</p> <p>Manifest.Resources.Resource.File {Href} 声明的值应是一个单一数字资源的定位器，此外没有其他含义。</p>

6.2.2.23 标识符

标识符 (Identifier) 的定义见表25。

表 25 Identifier 定义

描述符	定义/说明
组件名称	Identifier
组件类型	特性
数据类型	绑定语言相关的本地唯一标识符 (LUID)
值空间	与绑定语言相关
多重性	1..1
父辈	Manifest Organization Item Resource
描述	<p>Identifier 对象唯一标识 Manifest 对象中 Identifier 对象所属的元素。</p> <p>Identifier 的值可由使用 IdentifierRef 对象的其他对象进行内部引用。</p>

6.2.2.24 标识符引用

标识符引用 (IdentifierRef) 的定义见表26。

表 26 IdentifierRef 定义

描述符	定义/说明
组件名称	IdentifierRef
组件类型	特性
数据类型	绑定语言相关的引用现有LUID (LUIDref)
值空间	标识符引用对象中标识符对象的值包含内容清单对象，并受之后定义的内部引用规则的进一步约束
多重性	0..1: 作为Item的子辈 1..1: 作为Dependency的子辈
父辈	Item Dependency
描述	IdentifierRef 对象应准确复制一个 Identifier 对象的值。

6.2.2.25 可视性

可视性 (IsVisible) 的定义见表27。

表 27 IsVisible 定义

描述符	定义/说明
组件名称	IsVisible
组件类型	特性
数据类型	布尔值
值空间	真 假
多重性	0..1
父辈	Item
描述	<p>IsVisible 告知了这样一个渲染的过程：是否显示一个在同辈 Title 中声明的文本字符串，或者是否以其他任何方式可视化地指示 Item 对象的存在。</p> <p>该标签没有其他行为。对 IsVisible 所属的 Item 元素的任何后辈 Item，并不继承该 IsVisible 声明的可见度状态。</p> <p>IsVisible 的默认值为“真”，即使它未在 Item 的绑定实例中声明。也就是说，所属 Item 没有 IsVisible 相当于 Item 声明的 IsVisible 值为“真”。</p> <p>“真”值代表同辈 Title 的内容应由一个渲染的应用显示（即，Item{ IsVisible }.Title）。“假”值代表同辈 Title 的内容不应由一个渲染的应用显示（即，Item{ IsVisible }.Title）。</p>

6.2.2.26 语言

语言 (Language) 的定义见表28。

表 28 Language 定义

描述符	定义/说明
组件名称	Language
组件类型	特性
数据类型	字符串

表 28 (续)

描述符	定义/说明
值空间	遵循 GB 13000-2010 的规定
多重性	1
父辈	LingualTitle
描述	Language 对象标识包含在其所属 LingualTitle 对象中的字符串的人类语言。

6.2.2.27 其他

其他 (Other) 的定义见表29。

表 29 Other 定义

描述符	定义/说明
组件名称	Other
组件类型	特性
数据类型	本信息模型未知
值空间	本信息模型未知
多重性	0.. 不受限
父辈	Manifest Metadata Organizations Organization Item Resources Resource File Dependency
描述	<p>Other 对象是一个容器类组件中特征类的扩展。</p> <p>Other 对象是一个占位符。它将引入的特性类的有效位置告知这个信息模型的绑定，所引入的特性类是由本信息模型之外的一个信息模型定义的。</p> <p>一个扩展特性类的类型的真正名称，只有在组件的绑定被导入到信息模型的一个实例中时才会知道。任何扩展特征类的语义应在该信息模型中该类定义的父辈类的语义范围内。任何扩展特征类的语义不应覆盖或重新定义本信息模型中定义的任何父辈类的语义。</p> <p>Other 对象的范围受限於其所属元素。</p>

6.2.2.28 参数

参数 (Parameters) 的定义见表30。

表 30 Parameters 定义

描述符	定义/说明
组件名称	Parameters
组件类型	特性

表 30 (续)

描述符	定义/说明
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0..1
父辈	Item
描述	Parameters 对象提供一个位置用于声明某个静态、特定应用的信息，该信息始终与一个 Item 对象相关联。本信息模型不对参数进行定义。通常，参数由一个应用来定义并使用，该应用使用 Item 声明的信息，包括由 Parametes 所属的 Item 元素引用的任何 Resource 对象所指定的任何文件。 Parameters 的范围受限于其所属元素。

6.2.2.29 结构

结构 (Structure) 的定义见表31。

表 31 Structure 定义

描述符	定义/说明
组件名称	Structure
组件类型	特性
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	0..1
父辈	Organization
描述	Structure 对象描述一个 Organization 对象内 Item 对象彼此关联的特定方式。Structure 特性类的范围受限于其所属 Organization 元素。

6.2.2.30 类型

类型 (Type) 的定义见表32。

表 32 Type 定义

描述符	定义/说明
组件名称	Type
组件类型	特性
数据类型	字符串
值空间	见下列描述
多重性	1..1
父辈	Resource
描述	Type 对象提供一个术语、关键词或短语，指出由父辈 Resource 对象描述的资源类型。

6.2.2.31 版本

版本 (Version) 的定义见表33。

表 33 Version 定义

描述符	定义/说明
组件名称	Version
组件类型	特性
数据类型	字符串
值空间	遵循 GB 13000-2010 的规定
多重性	1..1
父辈	Resource
描述	Version 对象用一个值来指出一个 Manifest 对象的特定版本。如果作为 InterchangePackage 对象的子辈的 Manifest 声明此特性类，该值代表 Manifest 文档的版本。本标准未指定版本信息的语法。允许实现方使用满足他们需要的任何语法表示。Version 的范围受限于其所属元素。

7 内容包装

7.1 内容清单

7.1.1 内容清单文件

标记名：<manifest>。

在本标准中，所有内容都在Manifest文件中定义。

禁止使用Manifest的Version对象(例如version= ‘CP 1.0’)，以免与本标准的版本号混淆。而且，后者可隐含表示其使用的包装标准版本。

7.1.2 内容清单元数据

标记名：<manifestMetadata>。

ManifestMetadata元素以及它的Schema和SchemaVersion应在Manifest中定义，且使用如下格式：

```
<manifestMetadata >
  <schema>CCP</schema>
  <schemaVersion>1.0.0</schemaVersion>
```

... 依据通用包装的元数据 ...

```
</manifestMetadata >
```

在通用包装包的其他地方使用Metadata不受此限制。

通用包装在Manifest文件中提供了可选的角色元数据来指定相应Resource的权限。如不特别声明，则Resource默认向所有用户开放。如果特别声明，则可指定Resource对哪些角色可见。

能够访问Resource的角色在资源的元数据声明：

```
lom/educational/intendedEndUserRole/vocabulary/value
```

目前，默认的词汇表仅支持“学生”和“教师”两种角色。

7.1.3 组织结构

标记名：<organization>。

Organization用于描述通用包装包的文件夹内容类型。详见7.2中关于文件夹内容类型的讨论。

7.1.4 资源

标记名：<resource>。

通常，要描述包内的一段内容所需的全部额外数据将包含进一个单独文件并被Resource对象引用。

本标准使用Resource的Type特性类对象定义外部数据的类型。各种新的资源类型都会被逐个定义并纳入本标准。

如果需要使用多个不同Type的Resource文件来描述某个特定的内容实例时，不同Type的Resource应分别以<dependency>元素进行关联，而不应当捆绑在单独一个<resource>元素下。例如，一个QTI资源所需要的多个图像文件不应作为<file>元素直接引用，而应分别作为独立的类型为“Web内容”的Resource元素。

7.2 文件夹内容类型

7.2.1 类型说明

文件夹内容类型是一种基于结构/表现形式的内容组织形式。

通用包装包中的文件夹在某种意义上不同于Windows系统文件夹的包含关系，后者的文件夹包含所有内容，删除文件夹将删除它包含的全部内容。

不可通过文件路径中的文件夹名称直接引用该文件夹中的内容。

单个学习应用对象可被多次引用（既可以在一个文件夹内引用，也可以跨文件夹引用）。如果特定LMS不支持这种多次引用，则可通过生成该学习应用对象的多个副本实现。

7.2.2 组织结构集对象的用法

标记名：<organizations>。

文件夹内容类型由Organizations容器类型所表达，一个通用包装包可包含一个Organization，或不包含Organization。

Organizations的Default属性在通用包装包中无意义，故不允许使用。

7.2.3 组织结构对象的用法

标记名：<organization>。

Organization对象应包含值为“rooted-hierarchy”的结构属性。“rooted-hierarchy”为根层次结构，在面向对象编程中（但不是全部）最为典型。在大多数编程语言里，事实上，所有类直接或间接继承一个Root，然后形成一个共同继承层次结构。

Organization对象可包含一个Title对象值。该对象只在LMS如何表达组织结构时使用。例如，若LMS以学习模块的形式表达Organization，那么Organization的Title则可以作为模块的标题使用；如果LMS用现有课程下的一系列文件夹表达Organization，则将不使用Title属性。

```
<organizations/>
```

或者

```
<organizations>
```

```
  <organization identifier="Org1" structure="rooted-hierarchy">
```

```
    <title>数学</title>
```

```
    <item>...</item>
```

```
  </organization>
```

```
</organizations>
```

7.2.4 根文件夹

标记名: <root>。

包含文件夹组织结构的通用安装包只能使用一个Item元素为Root, Organization中不允许包含两个同辈的Item元素。作为Root的Item元素仅代表组织结构树的Root节点, 并无其他语义和指称, 而且不可包含Title元素。以下写法是正确的:

```
<organization identifier="Org1" structure="rooted-hierarchy">
  <item>
    <item>
      ...
    </item>
  </item>
</organization>
```

以下写法则是错误的:

```
<organization>
  <item>
    ...
  </item>
  <item>
    ...
  </item>
</organization>
或
<organization>
  <item>
    <title>某些文本</title>
  </item>
</organization>
```

7.2.5 内容项的用法

标记名: <item>。

Item容器对象类型代表一个文件夹或由文件夹指向的学习应用对象资源的链接。

本标准不支持Item元素的Parameters属性。

除作为Root的Item外, 每个Item都应包含Title元素。

7.2.6 表征文件夹的内容项

表征一个文件夹的Item对象不由IdentifierRef属性对象指示。

文件夹Item支持对其他文件夹Item以及学习应用对象链接的无限嵌套。

例如:

```
<item>
  <item>
    <title>根文件夹</title>
    <item>
      <title>子文件夹1</title>
    </item>
```

```

    <item>
      <title>子文件夹2</title>
      <item>
        <title>子文件夹2-子文件夹1</title>
      </item>
    </item>
  </item>
</item>

```

7.2.7 表征链接到学习应用对象的内容项

表征链接到学习应用对象的Item应具有IdentifierRef属性,指向描述被链接内容的Resource对象。学习应用对象的Item可被文件夹Item嵌套,但不应继续嵌套其他文件夹或者学习应用对象的Item。两个学习应用对象的Item可以引用相同的资源。

7.3 学习应用对象

每个学习应用对象应在内容清单中对应一个Resource元素。学习应用对象的实例包括QTI评估和论坛等。Resource元素的Type属性由学习应用对象表述的类型决定。如果除学习应用对象的描述符文件之外的其他文件存在于该对象所在目录或其中任何子目录下,这些文件在学习应用对象中的类型为“关联内容”Resource的Dependency元素下列出。

表述一个学习应用对象的资源应满足以下约束条件:

- 应包含一个File元素,该元素指向学习应用对象的描述符文件,即描述符文件与一个学习应用对象通过File元素建立关联;
- 不应包含其他任何File元素;
- 如果其他文件存在于学习应用对象的描述符文件所在目录或其中任何子目录下,这些文件资源应包含一个Dependency元素,该元素引用类型为“关联内容”的Resource;
- 不应包含任何其他Dependency元素来引用类型为“关联内容”的Resource;
- 可以包含任意个Dependency元素来引用类型为“Web内容”的Resource。

7.4 Web内容类型

类型为“Web内容”的资源应遵循以下约束条件:

- 针对位于包内的任何一个文件,应包含一个File元素,只要该文件不在一个学习应用对象的目录或其子目录下;
- 可以包含Dependency元素来引用任何类型为“Web内容”的其他资源;
- 不应包含Dependency元素来引用任何类型不是“Web内容”的其他资源。

7.5 包级Web内容类型

包级Web内容表达可被包装包内任何学习应用对象引用的Web内容。

包级Web内容被表征为一个Resource对象。

包级Web内容可以被一个文件夹的Item对象所引用。

包级Web内容的特性类Type应为“Web内容”。

如果从一个学习应用对象的Item对象处链接了包级Web内容资源,则该资源应包含Href特性类以表达可运行的资源。

引用来自学习应用对象文件夹中的文件时应使用相对路径,如:

“../filename”

7.6 关联内容类型

关联内容是一种Resource对象，是范围为特定Resource的Web内容。

它可以直接从文件夹Item对象引用。

其Type属性可以是“associatedcontent/cc_xmlv1p0/learning-application-resource”。

如果从一个学习应用对象的Item对象处链接了“关联内容”类型Resource，则该资源应包含Href特性类以表达可运行的资源。

该Resource对象可包含Dependency对象，后者引用了类型为“Web内容”的Resource对象。

类型为“关联内容”的资源应符合以下约束条件：

- a) 针对位于相关学习应用对象的描述符文件所在目录或其子目录中的每个文件，应包含一个File元素；
- b) 针对位于相关学习应用对象的描述符文件所在目录的上一级目录中的文件，不应包含对这些文件的任何引用；
- c) 不应包含任何Dependency元素。

7.7 讨论主题内容类型

标记名：<topic>。

讨论主题是用于开启讨论活动的学习应用对象。

讨论主题内容表示为Resource对象。

预期的默认行为：

- 导入后，讨论主题内容将被工具使用其自身表示方式存储；
 - 当安装包被添加到实际课程时，将通过工具在讨论论坛中创建相关讨论主题。
 - 包Organization中讨论主题Resource对象的位置决定了学习者被导向到论坛项目所在的课程中的位置。
 - 讨论主题小组将用于课程注册的学习者和导师间的合作。
 - 如果安装包被添加到多个课程中，每个课程将创建自己的讨论主题，且拥有自己的学习者小组。
- 讨论主题可直接被文件夹Item引用，Type属性值应为“dt_xmlv1p0”。

Resource对象的Href属性不可用。

Resource对象应包含一个File对象，该File对象引用讨论主题描述符XML文件。

Resource对象可包含Dependency对象，该对象引用“Web内容”和/或“关联内容”类型的Resource对象。

7.8 Web 链接（URL）内容类型

标记名：<webLink>。

Web链接内容类型可直接被文件夹Item对象引用。类型属性值为“wl_xmlv1p0”。

该Resource对象的Href属性不可用。

该Resource对象应包含一个File对象指向Web链接描述符XML文件的引用。

该Resource对象不可包含Dependency对象作为子对象。

Web链接在通用安装包的描述符文件中定义。

7.9 评估内容类型

标记名：<assessment>。

评估表现为Resource对象，可直接被文件夹Item对象引用，类型属性值应为“assessment”。
该Resource对象的Href属性不可用。

该Resource对象应包含一个File对象指向评估XML文件的引用。

该Resource对象可包含Dependency对象，但只能引用“Web内容”类型和/或“associatedcontent/cc_xmlvlp0/learning-application-resource”类型的资源对象。

7.10 题库内容类型

题库表征为Resource对象。文件夹Item对象不可直接引用题库。

如果通用安装包中包含题库，应在Manifest中作为一项Resource。但不可被包含在Organization内，并且任何学习应用对象都不可直接引用题库或题库中的题项。

仅允许教师查看题库，这时题库作为定制评估是使用的素材。LMS如何为教师提供题库功能不在本标准中讨论。

题库的类型属性值应是“question-bank”。

该Resource对象的Href属性不可用。

该Resource对象应包含一个指向题库XML文件的引用。

该Resource对象可包含Resource对象，但只能引用“Web内容”类型和/或“webcontent/cc_xmlvlp0/learning-application-resource”类型的Resource对象。

7.11 通用安装包权限

本标准支持两级授权，无论是整个通用安装包还是个别Resource都可以得到保护。这两个级别授权信息在通用包装包的Manifest中指定说明。

所有的权限均通过“授权”对象说明，它是Manifest中的扩展部分。如果授权对象不存在，就没有权限要求；如果授权对象存在，应首先定义通用包装包的权限，然后才是其他的权限机制。

如果是对通用安装包内的个别Resource而非整个通用安装包授权，那么可以利用保护属性“Protected”说明，它是Resource对象的扩展特性。

导入权限只在通用安装包级别可用。

如果指定了导入权限但未授权，那么通用安装包中的Resource和File均不可导入。

访问权限仅适用于包内Resource。

如果在包一级指定了访问权限，那么它将级联到Resource而非这些Resource包含的文件。

如果访问权限是针对某一个Resource，那么只能访问该Resource本身，不包括对Resource中包含的File访问。

通用安装包导入后的Resource管理不在本标准中讨论，授权只适用于访问通用安装包内Resource。

7.12 Web内容资源的路径

7.12.1 内容引用方法

为便于管理Web内容资源，特别是在内容导入到LMS系统后使用标准的HTML相对路径引用Resource，本标准设定Web内容资源在两个文件系统中使用：一是Resource使用的Web内容文件夹；二是学习应用对象的关联内容文件夹。

图4显示了如何跨不同文件系统通过相对路径引用Web内容。

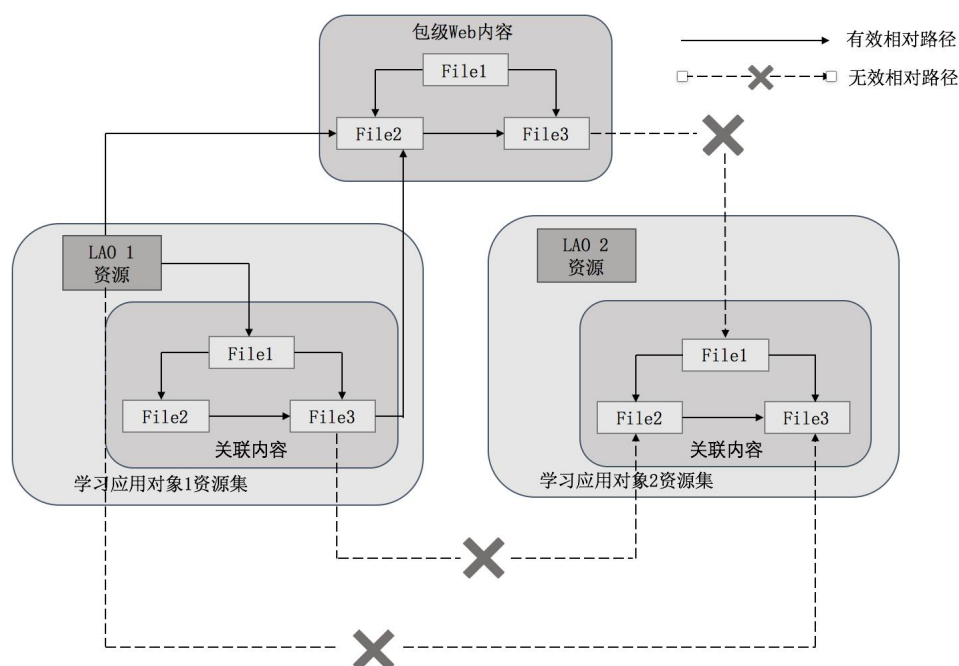


图 4 使用相对路径的内容引用

7.12.2 包级 Web 内容

如果包含包级Web内容文件夹，包级Web内容文件夹应出现在Root文件夹中。这个文件夹的名称不在本标准中定义。

包级Web内容文件夹下的Web内容可以被包内任何学习应用对象引用。本标准允许该文件夹和其子文件夹下文件之间的相对路径引用，但不允许该文件夹下的文件引用学习应用对象文件夹或其关联内容文件夹中的文件。

通常使用工具将这些内容导入到课程级的文件系统内。

7.12.3 学习应用对象的 Web 内容

每个通用安装包可包含1到n个学习应用对象的Web内容文件夹。

每个学习应用对象具有自己的关联内容文件系统文件夹。此文件夹及其子文件夹中的文件可引用此关联内容文件系统文件夹中的其他文件，以及使用相对路径语义的通用包装包的包级Web内容文件夹下的文件。但不应引用其他学习应用对象的Web内容文件夹或其子文件夹下的文件。此外，学习应用对象的Resource（如一个评估XML文件）还可包含对学习应用对象的关联内容文件夹及其子文件夹下的文件的引用，以及通用包装包的包级Web内容文件夹及其子文件夹下的文件的引用，但不应包含对其他学习应用对象文件夹中的Web内容的引用。

在导入方面，仅支持单一课程级文件系统的工具可以向课程级文件系统导入此Web内容。在此情况下，通用安装包中的Web内容表述一个较大的文件系统范围。

支持课程级和学习应用对象级文件系统的工具向仅供关联的学习应用对象使用的局部范围的File系统导入此类内容。

7.12.4 Web 内容文件系统内相对路径引用的方法

7.12.4.1 引用路径

在一个给定的文件系统（通用安装包或者学习应用对象）中，所有内容间的HTML链接（如，）均应使用相对路径。即无论该文件本身是Web 内容类型或者其他类型的资源文件（如评估文件）引用的文件路径是相对的。

7.12.4.2 从其他 Web 内容引用 Web 内容

在既有的文件系统中，路径和文件的位置有关。例如，对于以下文件：

```
content1/material/lesson.html
```

```
content1/images/icon.gif
```

lesson.html页面中对icon.gif的引用应采用如下格式：

```

```

7.12.4.3 从其他资源直接引用 Web 内容

当学习应用对象使用URI时，对于包含引用的文件应使用相对路径。

例如，对于以下文件：

```
content1/question1Qti.xml
```

```
content1/images/icon.gif
```

QTI mattext 元素的形式是：

```
<matimage uri="images/icon.gif">icon.gif</matimage>
```

7.12.4.4 从另一资源中的嵌入文本引用 Web 内容

学习应用对象的Resource支持包含HTML标签的文本，路径应和包含引用的文件对应。此外，还应包含特殊的令牌，使得在导入通用安装包的系统里查找和解析这些路径更为容易。

例如，对于以下File：

```
content1/question1Qti.xml
```

```
content1/images/icon.gif
```

引用嵌入到题项内容中的ICON图像应使用以下格式：

```

```

记号\$CESI--FILEBASE\$ 仅是一个便于发现路径的标志，它不替代其他记号。不过导入工具可以选择存储所引用的不同位置的文件，为了用工具处理路径事务可以随意插入任何所需的路径元素。可在资源里结合描述符文件使用\$CESI--FILEBASE\$记号。记号总是指向学习资源对象的基本目录，即，包含描述符的文件夹。遵循记号的路径应相对于 Resource 基本目录。因为在 LMS 中组织小测验或讨论会用的 Dependency 文件时使用的方法可能打断嵌入 Resource 的连接，所以，LMS 在导入时或导入后应换一个记号使用另一条路径。此外，对于需要的任何补充图像或 HTML，应按 Dependency 关系列出。

7.12.5 学习应用对象指向包内文件系统的相对路径引用格式

7.12.5.1 引用路径

从学习应用对象中的内容指向通用安装包中Web内容的HTML链接（如，）应使用相对路径来定义。在此，相对路径是指File元素包含的引用关系，无论File元素本身是Web内容或者其他资源文件（如评估文件）。

7.12.5.2 从学习应用对象的 Web 内容引用包级 Web 内容

若学习应用对象的文件系统中有一条相对路径指向该文件系统Root目录的上级目录,那么该引用指向的是通用包装包的Web内容的文件系统。例如,对于以下File:

```
images/icon.gif
  content1/material/lesson.html
lesson.html中对icon.gif图像引用应使用以下路径:

```

7.12.5.3 从学习应用对象资源直接引用包级 Web 内容

当另一个学习应用对象的Resource中通过URI引用Web Content时,应使用包含该引用的那个File的相对路径。

例如,对于以下File:

```
images/icon.gif
  content1/question1Qti.xml
QTI matimage元素应以以下格式引用包装包中的图标:
<matimage uri="../../images/icon.gif">icon.gif</matimage>
```

7.12.5.4 从另一资源中的嵌入文本引用包级 Web 内容

学习应用对象的Resource支持包含HTML标签的文本,路径应和包含引用的File对应。此外还应包含一个特殊的令牌,以便在导入通用包装包的系统中查找和解析这些路径。

例如,对于以下File:

```
images/icon.gif
  content1/question1Qti.xml
在题目内容中引用的通用包中图标File,应使用以下格式:
。
```

7.12.6 Dependency 元素

Dependency元素只能用于特定类型的资源以及资源类型之间的某些关系。规则如下:

Web 链接——不应有 Dependency 元素;

讨论主题——可以包含关于 Web 内容资源和/或关联内容资源的 Dependency;

评估——可以包含关于 Web 内容资源和/或关联内容资源的 Dependency;

问题库——可以包含关于 Web 内容资源和/或关联内容资源的 Dependency;

Web 内容——可以包含关于 Web 内容资源的 Dependency;

关联内容——可以包含关于 Web 内容资源的 Dependency;

LTI——可以仅包含关于图像图标的 Dependency;

讨论主题、Web 链接、评估和题库资源不应被 Dependency 标识;

Dependency不应指示自己的父辈资源元素,并且资源的每个Dependency应指向唯一资源,即Dependency子元素不应被父辈资源重复使用。

参 考 文 献

- [1]GB/T 21365-2008 信息技术 学习、教育和培训 学习对象元数据
 - [2]GB/T 26222-2010 信息技术 学习、教育和培训 内容包装
 - [3]GB/T 29802-2013 信息技术 学习、教育和培训 测试试题信息模型
-