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

**Part 2:
XML binding**

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

Partie 2: Liaison XML



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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 |
| Introduction..... | v |
| 1 Scope | 1 |
| 2 Normative references | 1 |
| 3 Terms and definitions | 1 |
| 4 Abbreviated terms | 3 |
| 5 XML schema documentation..... | 4 |
| 5.1 Core content packaging binding description | 4 |
| 5.1.1 Schema document properties | 4 |
| 5.1.2 Global declarations | 5 |
| 5.1.2.1 Element: manifest | 5 |
| 5.1.3 Global definitions | 6 |
| 5.1.3.1 Complex Type: Dependency.Type | 6 |
| 5.1.3.2 Complex Type: File.Type | 7 |
| 5.1.3.3 Complex Type: Item.Type | 8 |
| 5.1.3.4 Complex Type: Manifest.Type | 10 |
| 5.1.3.5 Complex Type: ManifestMetadata.Type | 11 |
| 5.1.3.6 Complex Type: Metadata.Type | 12 |
| 5.1.3.7 Complex Type: Organization.Type..... | 13 |
| 5.1.3.8 Complex Type: Organizations.Type..... | 15 |
| 5.1.3.9 Complex Type: Resource.Type | 16 |
| 5.1.3.10 Complex Type: Resources.Type | 17 |
| 5.1.3.11 Model Group: grpStrict.any | 18 |
| 5.2 Content packaging extension binding description..... | 19 |
| 5.2.1 Schema document properties | 19 |
| 5.2.2 Global declarations | 20 |
| 5.2.2.1 Element: ipointer..... | 20 |
| 5.2.2.2 Element: lingualTitle..... | 21 |
| 5.2.2.3 Element: variant..... | 21 |
| 5.2.3 Global definitions | 22 |
| 5.2.3.1 Complex Type: IPointer.Type | 22 |
| 5.2.3.2 Complex Type: LingualTitle.Type | 24 |
| 5.2.3.3 Complex Type: Metadata.Type | 24 |
| 5.2.3.4 Complex Type: Variant.Type | 25 |
| 5.2.3.5 Model Group: grpStrict.any | 26 |
| 6 Using schematron | 27 |
| Annex A (informative) Document Provenance..... | 28 |
| Annex B (informative) Listing of the XSDs..... | 29 |
| Bibliography..... | 30 |

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.

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

ISO/IEC 12785-2 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*

Introduction

This part of ISO/IEC 12785 will be used as the basis for the production of the following documents:

- ISO/IEC 12785-3 (*Best practice and implementation guide*);
- Content Packaging XML XSD.¹⁾

This part of ISO/IEC 12785 details how the ISO/IEC 12785-1 information model should be represented using XML schema. The content packaging binding is contained in two XML schemas and two vocabulary files. The vocabulary files are instances of the IMS Vocabulary Definition Exchange specification. ISO/IEC 12785-3 provides non-normative guidance on how to use the binding and information model. For a conceptual overview of ISO/IEC 12785, see ISO/IEC 12785-1. For a discussion of potential applications, see ISO/IEC 12785-3. Conformance to ISO/IEC 12785 is addressed within ISO/IEC 12785-1.

1) This XML schema definition can be accessed from: <http://www.imsglobal.org/content/packaging/index.html>.

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

Part 2: XML binding

1 Scope

This part of ISO/IEC 12785 specifies how to represent the ISO/IEC 12785-1 information model in XML, and details each element binding of the content packaging XML schema.

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 references, 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*

3 Terms and definitions

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

3.1

binding

XML data binding

means of representing the information in an XML document

NOTE See XML data binding of Wikipedia: http://en.wikipedia.org/wiki/XML_data_binding (retrieved November 18, 2009).

3.2

control file

single computer file that governs the binding of the Content Packaging Information Model 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.

EXAMPLE A file containing an XML schema can be used as a control file for an XML binding of a manifest.

3.3

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 logical package can contain one or more units of content.

3.4

logical package

representation of one or more units of usable (and reusable) content

NOTE A logical package encompasses the full set of components described by the manifest and its child manifests, including the local components and the remote components included by reference.

[ISO/IEC 12785-1:2009]

3.5

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]

3.6

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 core structures 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]

3.8

organization

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

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

3.9

package

unit of usable (and reusable) content

NOTE 1 This can be part of a learning course that has instructional relevance outside of a 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.

3.10

resource

⟨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.

3.11**schema****XML Schema**

description of a class of XML documents, expressed in terms of constraints on the structure and content of those documents

NOTE 1 For more information see <http://www.w3c.org/XML/Schema>.

NOTE 2 XML Schema in Content Packaging follows W3C recommendation *XML Schema*.

NOTE 3 The definition refers to the concept of a schema in an XML context, as well as a specific language for creating such schemas: the W3C XML Schema. The "IMS Content Packaging XML XSD" is an application of the W3C XML Schema language.

3.12**unit of content**

file or grouping of files which can be represented within a manifest

3.13**XML validation**

process whereby documents written in XML (eXtensible Markup Language) are verified against the defined structure

NOTE Strict validation in XML means that it must follow the rules dictated by an XML schema.

4 Abbreviated terms

| | |
|-------|---------------------------------------|
| CPIM | Content Packaging Information Model |
| I-BAT | IMS Binding Auto-generation Tool-kit |
| LET | Learning, Education, and Training |
| MDA | Model Driven Architecture |
| PIM | Protocol Independent Model |
| PSM | Platform Specific Model |
| UML | Unified Modeling Language |
| VDEX | Vocabulary Definition Exchange |
| W3C | World Wide Web Consortium |
| XMI | XML Metadata Interchange |
| XML | eXtensible Mark-up Language (W3C XML) |
| XSD | XML Schema Definition |
| XSL | Extensible Stylesheet Language |
| XSLT | XSL Transformations |

5 XML schema documentation

5.1 Core content packaging binding description

5.1.1 Schema document properties

The following subclauses outline the characteristics of each element of the content packaging XML schema binding for core²⁾ elements in tabular form. The same information is also presented as an XSD in Annex B.1.

In accordance with the IMS Global Learning Consortium (GLC) Namespace Policy [IMS-NAMESPACE] specified in IMS Content Packaging XML Binding, the “name” of an element is appended to a IMS GLC namespace URI to construct a Uniform Resource Identifier as a globally unique identifier for that element. The use of element names and URIs in the context of different implementation technologies is explained in ISO/IEC 12785-2.

To describe XML instance representation of each element declared namespaces are as follows:




| Prefix | Namespaces |
|-------------------|---|
| Default namespace | http://www.imsglobal.org/xsd/imscp_v1p1 |
| xml | http://www.w3.org/XML/1998/namespace |
| xs | http://www.w3.org/2001/XMLSchema |
| xsi | http://www.w3.org/2001/XMLSchema-instance |

Schema component representation:



```
<xs:schema targetNamespace="http://www.imsglobal.org/xsd/imscp_v1p2" version="IMS CP 1.2"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
```

```
...
</xs:schema>
```

The following subclauses was created using the schema documentation tool provided as part of the Oxygen product. To describe logical diagram used notations are as follows:

- @ : xs:attribute
-  : xs:sequence
-  : xs:anyAttribute
-  : xs:any namespace

2) The core refers to all those elements that were used in the IMS Content Packaging specification versions prior to version 1.2 which is source of ISO/IEC 12785.

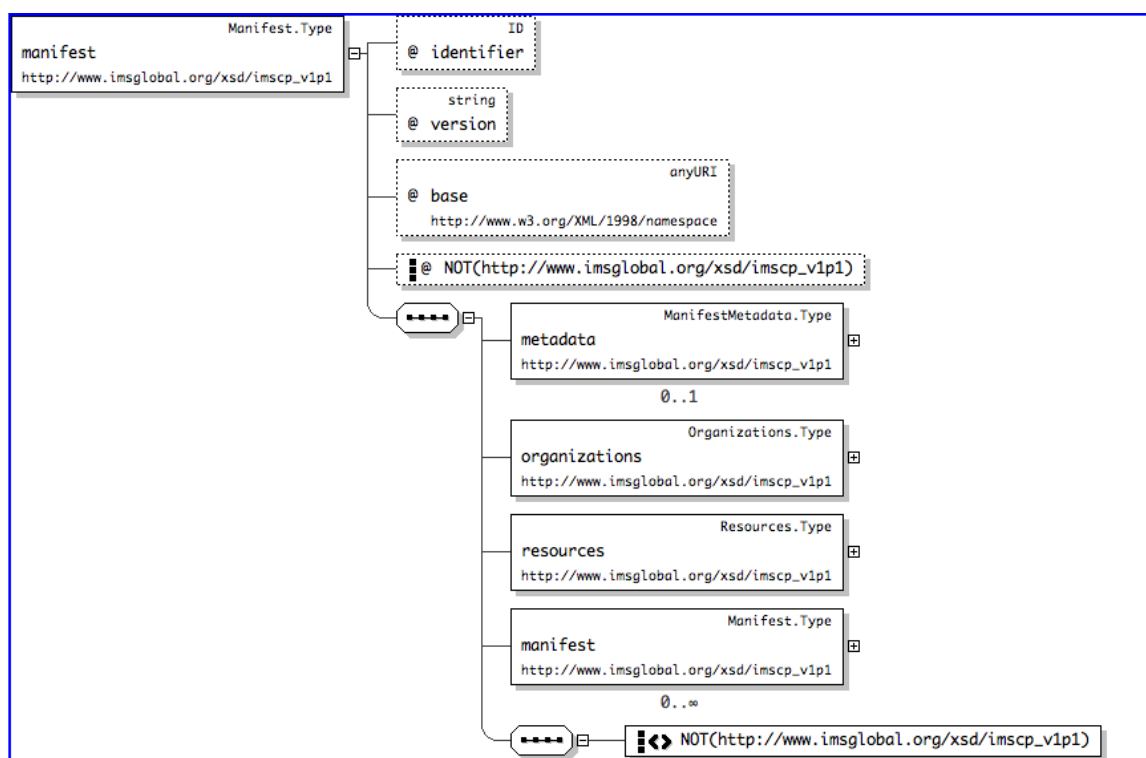
-  : xs:group
- 0..1 : property for zero to one
- 0..∞ : property for zero to unbounded
-  : xs:complexType

5.1.2 Global declarations

5.1.2.1 Element: manifest

| | |
|-----------------|---------------|
| Name | manifest |
| Type | Manifest.Type |
| Nilable | no |
| Abstract | no |

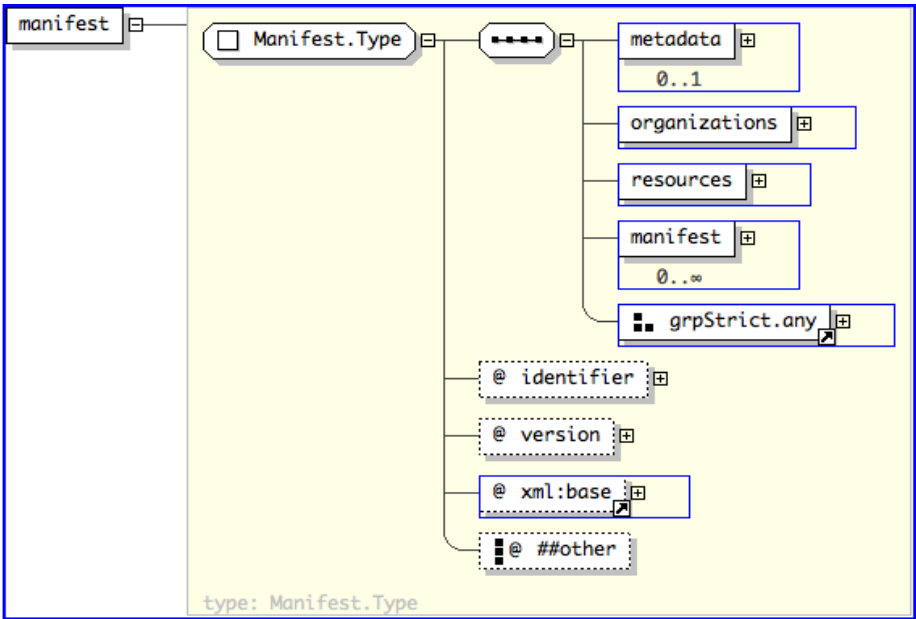
Logical diagram of manifest:



XML instance representation:

```
<manifest
  identifier=" xs:ID [1]"
  version=" xs:string [0..1]"
  xml:base="[0..1]"
  Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <metadata> ManifestMetadata.Type </metadata> [0..1]
  <organizations> Organizations.Type </organizations> [1]
  <resources> Resources.Type </resources> [1]
  <manifest> Manifest.Type </manifest> [0..*]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</manifest>
```

Diagram:



Schema component representation:

```
<xs:element name="manifest" type=" Manifest.Type "/>
```

5.1.3 Global definitions

5.1.3.1 Complex Type: Dependency.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|---|----------------------------|
| Name | Dependency.Type |
| Used by (from the same schema document) | Complex Type Resource.Type |
| Abstract | no |

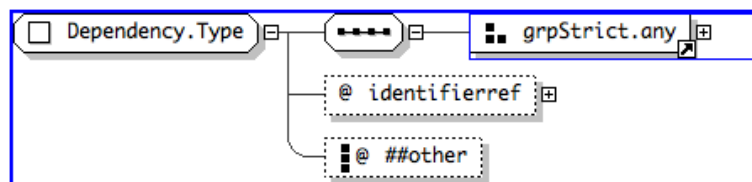
| | |
|----------------------|--|
| Documentation | <p>A dependency element provides a way to associate another collection of file references within the scope of the dependency element's parent resource element.</p> <p>Shared file references can be declared once and associated many times through a Dependency element.</p> <p>This element represents a binding of the kinds of child objects defined for Dependency.[Extension]³⁾ (see subclause 6.6.4 of ISO/IEC 12785-1:2009).</p> |
|----------------------|--|

XML instance representation:

```

<...
  identifierref=" xs:string [1]"
  Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="Dependency.Type">
  <xs:sequence>
    <xs:group ref=" grpStrict.any " />
  </xs:sequence>
  <xs:attribute name="identifierref" type=" xs:string " use="required"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

5.1.3.2 Complex Type: File.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|----------------------------|
| Name | File.Type |
| Used by (from the same schema document) | Complex Type Resource.Type |
| Abstract | no |

3) Symbol '[']' means children elements.

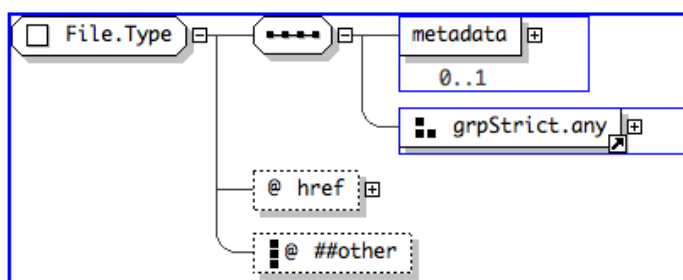
| | |
|----------------------|--|
| Documentation | <p>A file element declares a reference to a file. The reference may be relative to the Package containing the file element or absolute (external to the Package).</p> <p>A file element may contain child extensions declaring alternative references to the same asset⁴⁾ as that referenced by the file element's href attribute.</p> <p>This element represents a binding of the kinds of child objects defined for File.[Metadata, Extension] (see subclause 6.6.3 of ISO/IEC 12785-1:2009).</p> |
|----------------------|--|

XML instance representation:

```

<...
href=" xs:anyURI [1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <metadata> Metadata.Type </metadata> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="File.Type">
  <xs:sequence>
    <xs:element name="metadata" type=" Metadata.Type " minOccurs="0" maxOccurs="1"/>
    <xs:group ref=" grpStrict.any "/>
  </xs:sequence>
  <xs:attribute name="href" type=" xs:anyURI " use="required"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

5.1.3.3 Complex Type: Item.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

4) 'Asset' and 'asset references' refer to a variety of content packaging components that refer directly or indirectly to the files that a particular logical package aggregates.

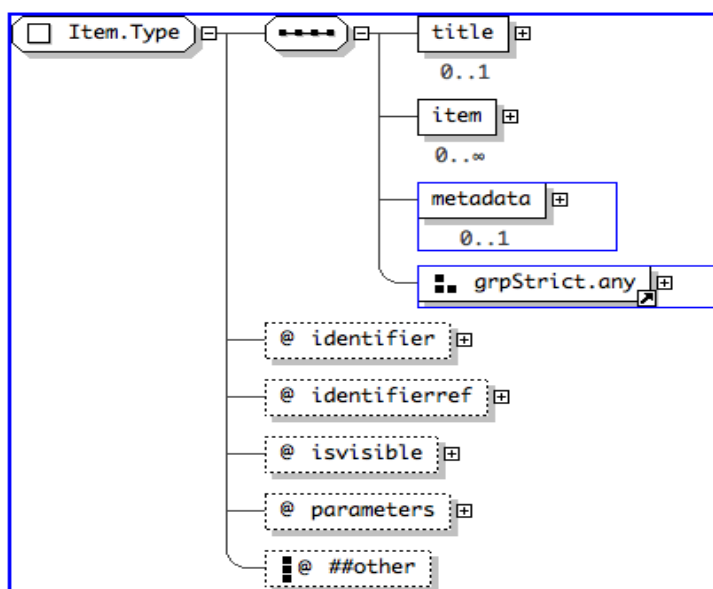
| | |
|--|--|
| Name | Item.Type |
| Used by (from the same schema document) | Complex Type Organization.Type, Complex Type Item.Type |
| Abstract | no |
| Documentation | <p>An item element represents a structural node in a particular organization. An item element may be a parent or sibling of other Item elements, each one representing a unique structural node.</p> <p>An organization has no purpose unless it has at least one Item element.</p> <p>This element represents a binding of the kinds of child objects of Item.[Title, Item, Metadata, Extension] (see subclause 6.5.5 of ISO/IEC 12785-1:2009).</p> |

XML instance representation:

```

<...
identifier=" xs:ID [1]"
identifierref=" xs:string [0..1]"
isvisible=" xs:boolean [0..1]"
parameters=" xs:string [0..1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <title> xs:string </title> [0..1]
  <item> Item.Type </item> [0..*]
  <metadata> Metadata.Type </metadata> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:

Schema component representation:

```

<xs:complexType name="Item.Type">
  <xs:sequence>
    <xs:element name="title" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="item" type="Item.Type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="metadata" type="Metadata.Type" minOccurs="0" maxOccurs="1"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
  <xs:attribute name="identifier" type="xs:ID" use="required"/>
  <xs:attribute name="identifierref" type="xs:string" use="optional"/>
  <xs:attribute name="isvisible" type="xs:boolean" use="optional"/>
  <xs:attribute name="parameters" type="xs:string" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

5.1.3.4 Complex Type: Manifest.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|--|
| Name | Manifest.Type |
| Used by (from the same schema document) | Complex Type Manifest.Type, Element manifest |
| Abstract | no |
| Documentation | <p>A manifest element is a container for data structures whose contents describe a valid instance of ISO/IEC 12785-1.</p> <p>A manifest element may contain and reference child manifest elements in the same manifest document. The root manifest element defines an entire content package. A child manifest element defines a semantically complete subset of that Package.</p> <p>This element represents a binding of the kinds of objects defined as corresponding child manifests. [ManifestMetadata, Organizations, Resources, Manifest, Extension] (see 6.4.1 of ISO/IEC 12785-1:2009).</p> |

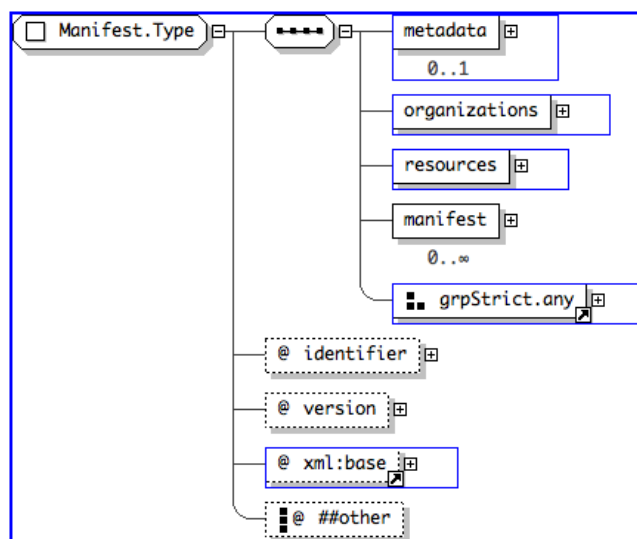
XML instance representation:

```

<...
identifier="xs:ID [1]"
version="xs:string [0..1]"
xml:base="[0..1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <metadata> ManifestMetadata.Type </metadata> [0..1]
  <organizations> Organizations.Type </organizations> [1]
  <resources> Resources.Type </resources> [1]
  <manifest> Manifest.Type </manifest> [0..*]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```


Diagram:



Schema component representation:

```

<xs:complexType name="Manifest.Type">
  <xs:sequence>
    <xs:element name="metadata" type="ManifestMetadata.Type" minOccurs="0" maxOccurs="1"/>
    <xs:element name="organizations" type="Organizations.Type" minOccurs="1" maxOccurs="1"/>
    <xs:element name="resources" type="Resources.Type" minOccurs="1" maxOccurs="1"/>
    <xs:element name="manifest" type="Manifest.Type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
  <xs:attribute name="identifier" type="xs:ID" use="required"/>
  <xs:attribute name="version" type="xs:string" use="optional"/>
  <xs:attribute ref="xml:base" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>
  
```

5.1.3.5 Complex Type: ManifestMetadata.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|----------------------------|
| Name | ManifestMetadata.Type |
| Used by (from the same schema document) | Complex Type Manifest.Type |
| Abstract | no |

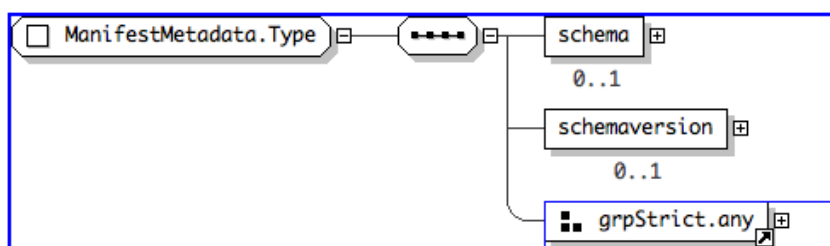
| | |
|----------------------|---|
| Documentation | <p>This metadata element contains data structures that declare descriptive information about an entire Package.</p> <p>One or more different metadata models may be declared as child extensions of a metadata element.</p> <p>The schema and schemaversion children define the kind or collection of metadata models being used.</p> <p>This element represents a binding of the kinds of child objects defined for ManifestMetadata.[Schema, SchemaVersion, MetadataModel] (see subclause 6.4.2 of ISO/IEC 12785-1:2009).</p> |
|----------------------|---|

XML instance representation:

```

<...>
  <schema> xs:string </schema> [0..1]
  <schemaversion> xs:string </schemaversion> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="ManifestMetadata.Type">
  <xs:sequence>
    <xs:element name="schema" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="schemaversion" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:group ref="grpStrict.any" />
  </xs:sequence>
</xs:complexType>

```

5.1.3.6 Complex Type: Metadata.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|--|
| Name | Metadata.Type |
| Used by (from the same schema document) | Complex Type Organization.Type, Complex Type Resource.Type, Complex Type Item.Type, Complex Type File.Type |
| Abstract | no |

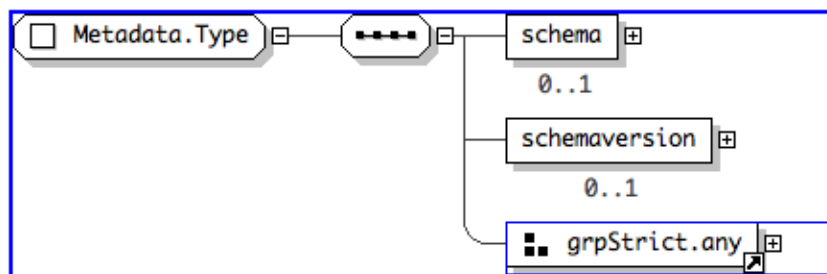
| | |
|----------------------|---|
| Documentation | <p>An instance of the metadata element contains data structures that declare descriptive information about a metadata element's parent only.</p> <p>One or more different metadata models may be declared as child extensions of a metadata element.</p> <p>This element represents a binding of the kinds of child objects defined for Metadata.[Extension] (see subclause 6.7 of ISO/IEC 12785-1:2009).</p> |
|----------------------|---|

XML instance representation:

```

<...>
  <schema> xs:string </schema> [0..1]
  <schemaversion> xs:string </schemaversion> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="Metadata.Type">
  <xs:sequence>
    <xs:element name="schema" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="schemaversion" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
</xs:complexType>

```

5.1.3.7 Complex Type: Organization.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|---|
| Name | Organization.Type |
| Used by (from the same schema document) | Complex Type Organizations.Type |
| Abstract | no |
| Documentation | An organization element serves as a container for all those data structures that describe a particular view on all the information encapsulated by the organization object's manifest grandparent object. |

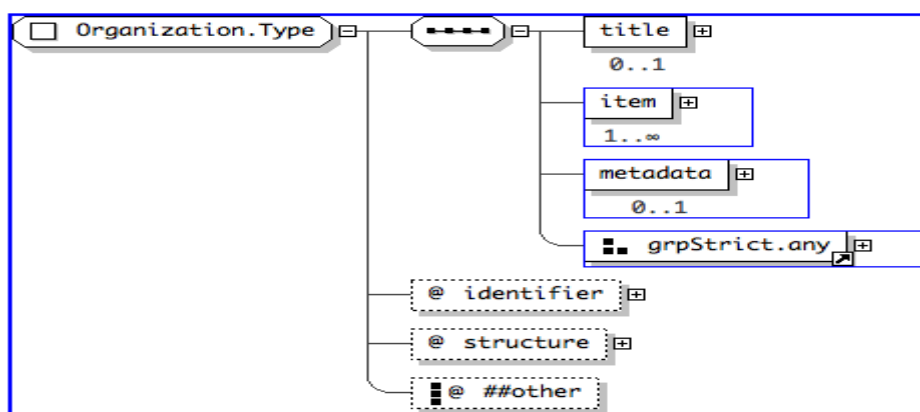
| | |
|--|---|
| | <p>Multiple organization elements within the same parent organizations element are equivalent in purpose: Each shows a different way for structuring the same information declared within a grandparent manifest object.</p> <p>This element represents a binding of the kinds of child objects defined for Organization[Title, Item, Metadata, Extension] (see subclause 6.5.2 of ISO/IEC 12785-1:2009).</p> |
|--|---|

XML instance representation:

```

<...
identifier=" xs:ID [1]"
structure=" xs:string [0..1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <title> xs:string </title> [0..1]
  <item> Item.Type </item> [1..*]
  <metadata> Metadata.Type </metadata> [0..1]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="Organization.Type">
  <xs:sequence>
    <xs:element name="title" type=" xs:string " minOccurs="0" maxOccurs="1"/>
    <xs:element name="item" type=" Item.Type " minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="metadata" type=" Metadata.Type " minOccurs="0" maxOccurs="1"/>
    <xs:group ref=" grpStrict.any "/>
  </xs:sequence>
  <xs:attribute name="identifier" type=" xs:ID " use="required"/>
  <xs:attribute name="structure" type=" xs:string " use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

5.1.3.8 Complex Type: Organizations.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|---|
| Name | Organizations.Type |
| Used by (from the same schema document) | Complex Type Manifest.Type |
| Abstract | no |
| Documentation | <p>The organizations element is a container for all data structures that describe the way or ways that information encapsulated by its parent manifest element is structured.</p> <p>This represents of binding of the child objects of Organizations. [Organization, Extension] (see 6.5.1 of ISO/IEC 12785-1:2009).</p> |

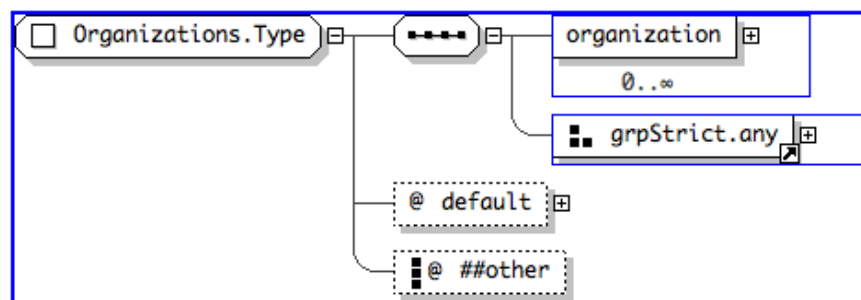
XML instance representation:

```

<...
default=" xs:IDREF [0..1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
    <organization> Organization.Type </organization> [0..*]
    Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:



Schema component representation:

```

<xs:complexType name="Organizations.Type">
  <xs:sequence>
    <xs:element name="organization" type="Organization.Type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
  <xs:attribute name="default" type="xs:IDREF" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

5.1.3.9 Complex Type: Resource.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|---|
| Name | Resource.Type |
| Used by (from the same schema document) | Complex Type Resources.Type |
| Abstract | no |
| Documentation | <p>A resource element is a container for a particular file or collection of files. A resource may contain references to files that are all of the same type or different types (i.e., file formats).</p> <p>This element represents a binding of the kinds of child objects defined for Resource.[Metadata, File, Dependency, Extension] (see subclause 6.6.2 of ISO/IEC 12785-1:2009).</p> |

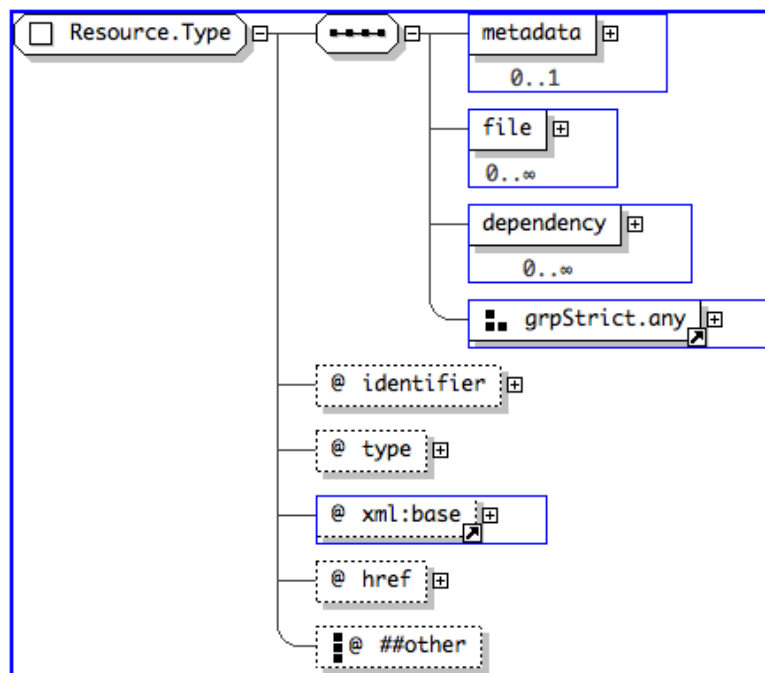
XML instance representation:

```

<...
  identifier=" xs:ID [1]"
  type=" xs:string [1]"
  xml:base="[0..1]"
  href=" xs:anyURI [0..1]"
  Allow any attributes from a namespace other than this schema's namespace (strict validation).
  >
    <metadata> Metadata.Type </metadata> [0..1]
    <file> File.Type </file> [0..*]
    <dependency> Dependency.Type </dependency> [0..*]
    Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
  </...>

```

Diagram:



Schema component representation:

```

<xs:complexType name="Resource.Type">
  <xs:sequence>
    <xs:element name="metadata" type="Metadata.Type" minOccurs="0" maxOccurs="1"/>
    <xs:element name="file" type="File.Type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="dependency" type="Dependency.Type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
  <xs:attribute name="identifier" type="xs:ID" use="required"/>
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute ref="xml:base" use="optional"/>
  <xs:attribute name="href" type="xs:anyURI" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>
  
```

5.1.3.10 Complex Type: Resources.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|----------------------------|
| Name | Resources.Type |
| Used by (from the same schema document) | Complex Type Manifest.Type |
| Abstract | no |

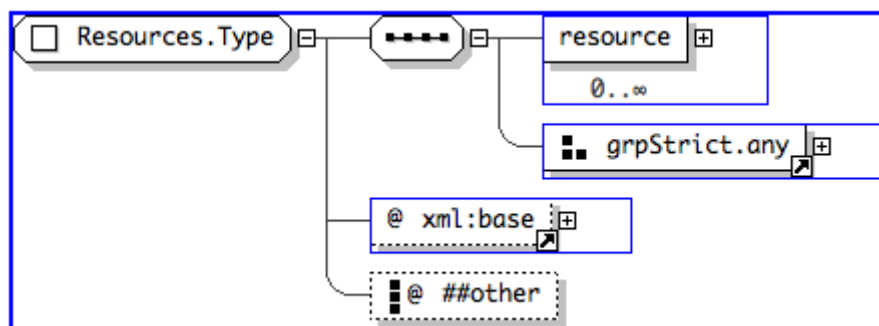
| | |
|----------------------|---|
| Documentation | <p>The Resources element is a container for data structures containing references to one or more files. File references may be grouped within a containing resources element in whatever manner seems most appropriate.</p> <p>The scope of referenced files is specific to a resources element's parent manifest element only.</p> <p>This element represents a binding of the kinds of child objects defined for Resources.[Resource, Extension] (see 6.6.1 of ISO/IEC 12785-1:2009).</p> |
|----------------------|---|

XML instance representation:

```

<...
xml:base="[0..1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  <resource> Resource.Type </resource> [0..*]
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="Resources.Type">
  <xs:sequence>
    <xs:element name="resource" type="Resource.Type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="grpStrict.any"/>
  </xs:sequence>
  <xs:attribute ref="xml:base" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

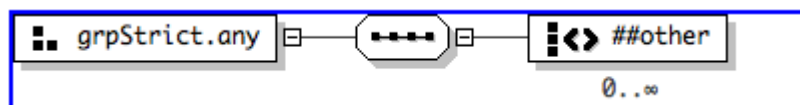
```

5.1.3.11 Model Group: grpStrict.any

| | |
|--|--|
| Name | grpStrict.any |
| Used by (from the same schema document) | Complex Type Manifest.Type, Complex Type Metadata.Type, Complex Type Organizations.Type, Complex Type Resources.Type, Complex Type Organization.Type, Complex Type Resource.Type, Complex Type Item.Type, Complex Type File.Type, Complex Type Dependency.Type, Complex Type ManifestMetadata.Type |
| Documentation | Any namespaced element from any namespace may be included within an "any" element. The namespace for any element that is defined in another standard or specification must be declared in the instance, and the schema must be referenced. The extension has a definition of "strict" i.e. each extension element must have its own namespace. |

XML instance representation:

Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]

Diagram:**Schema component representation:**

```
<xs:group name="grpStrict.any">
  <xs:sequence>
    <xs:any namespace="##other" processContents="strict" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>
```

5.2 Content packaging extension binding description**5.2.1 Schema document properties**

The following subclauses outline the characteristics of each element of the content packaging XML schema binding for extension⁵⁾ elements in tabular form. The same information is also presented as an XSD in Annex B.2.

In accordance with the IMS GLC Namespace Policy [IMS-NAMESPACE] specified in IMS Content Packaging XML Binding, the “name” of an element is appended to a IMS GLC namespace URI to construct a Uniform Resource Identifier as a globally unique identifier for that element. The use of element names and URIs in the context of different implementation technologies is explained in ISO/IEC 12785-2.

To describe XML instance representation of each element declared namespaces are as follows:

| Prefix | Namespaces |
|-------------------|--|
| Default namespace | http://www.imsglobal.org/xsd/imscp_extensionv1p2 |
| xml | http://www.w3.org/XML/1998/namespace |
| xs | http://www.w3.org/2001/XMLSchema |
| xsi | http://www.w3.org/2001/XMLSchema-instance |
| xlink | http://www.w3.org/1999/xlink |

Schema component representation:

```
<xs:schema targetNamespace="http://www.imsglobal.org/xsd/imscp_extensionv1p2" version="IMS CPX 1.2"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://www.imsglobal.org/xsd/ims_xlink.xsd"/>
  ...
</xs:schema>
```

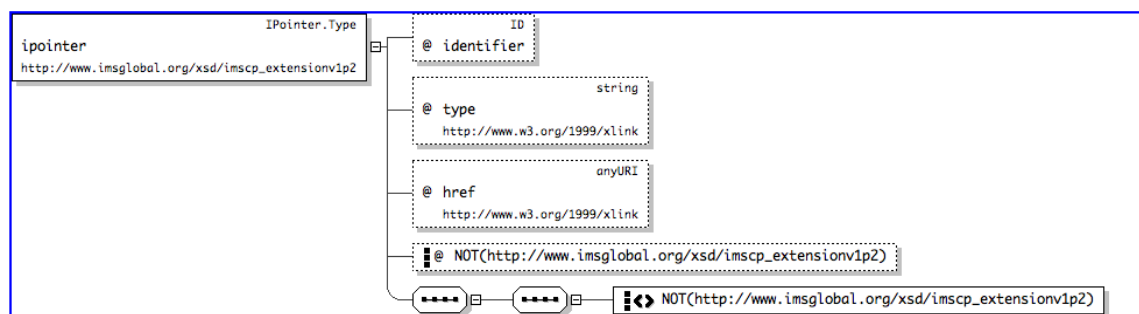
5) The extension refers to those elements that have been newly introduced with IMS Content Packaging version 1.2 which is source of ISO/IEC 12785. These are defined in a unique namespace that is not shared with any other version of IMS Content Packaging specification.

5.2.2 Global declarations

5.2.2.1 Element: ipointer

| | |
|-----------------|---------------|
| Name | ipointer |
| Type | IPointer.Type |
| Nilable | no |
| Abstract | no |

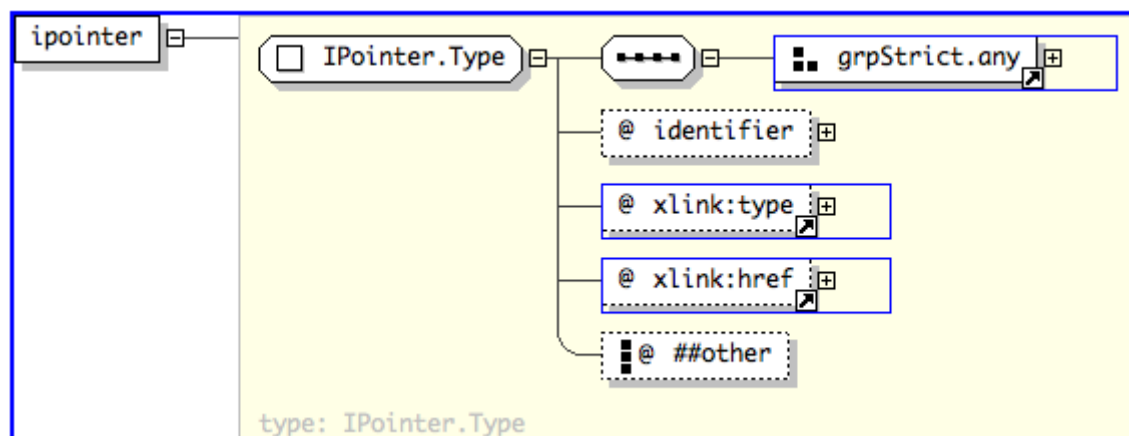
Logical diagram of ipointer:



XML instance representation:

```
<ipointer
  identifier="xs:ID [1]"
  xlink:type="[0..1]"
  xlink:href="[1]"
  Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
  Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</ipointer>
```

Diagram:



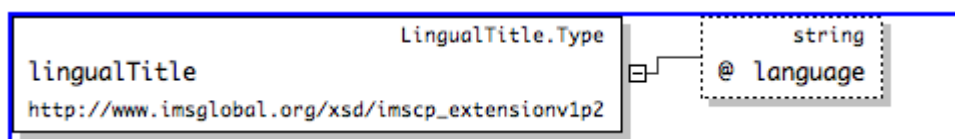
Schema component representation:

```
<xs:element name="ipointer" type="IPointer.Type"/>
```

5.2.2.2 Element: lingualTitle

| | |
|-----------------|-------------------|
| Name | lingualTitle |
| Type | LingualTitle.Type |
| Nilable | no |
| Abstract | no |

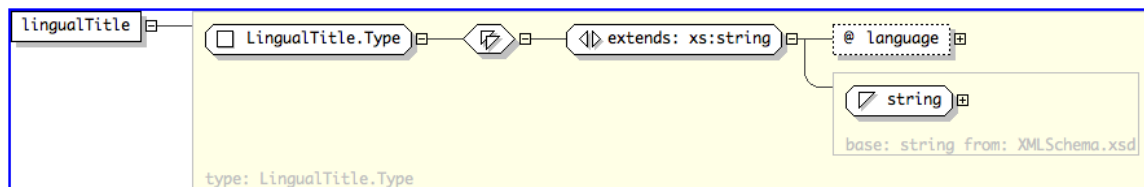
Logical diagram:



XML instance representation:

```
<lingualTitle
  language=" xs:string [1]">
  xs:string
</lingualTitle>
```

Diagram:



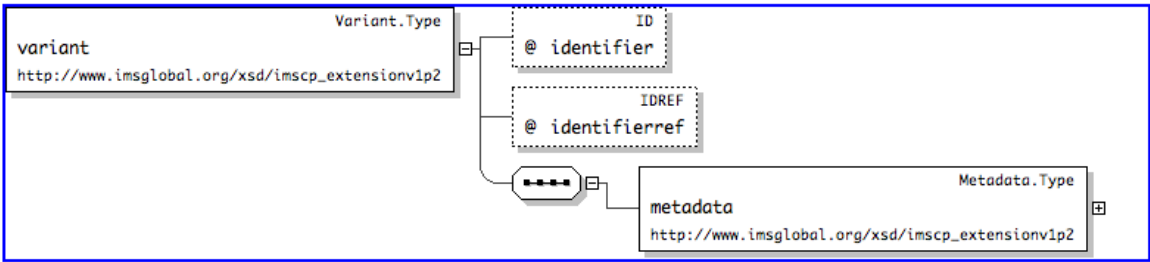
Schema component representation:

```
<xs:element name="lingualTitle" type=" LingualTitle.Type "/>
```

5.2.2.3 Element: variant

| | |
|-----------------|--------------|
| Name | variant |
| Type | Variant.Type |
| Nilable | no |
| Abstract | no |

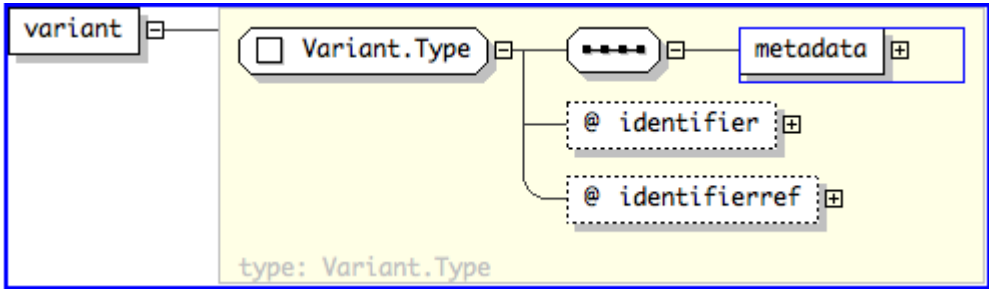
Logical diagram:



XML instance representation:

```
<variant
  identifier="xs:ID [1]"
  identifierref="xs>IDREF [1]">
  <metadata> Metadata.Type </metadata> [1]
</variant>
```

Diagram:



Schema component representation:

```
<xs:element name="variant" type="Variant.Type"/>
```

5.2.3 Global definitions

5.2.3.1 Complex Type: IPointer.Type

| | |
|---|------------------|
| Super-types: | None |
| Sub-types: | None |
| Name | IPointer.Type |
| Used by (from the same schema document) | Element ipointer |
| Abstract | no |

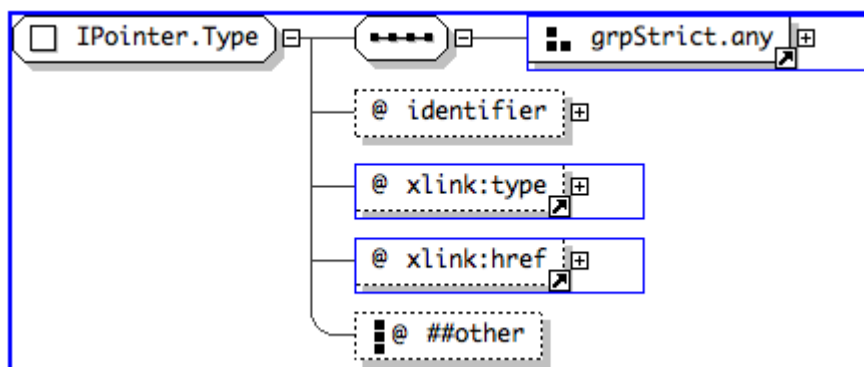
| | |
|----------------------|--|
| Documentation | <p>A bound instance of an IPointer object allows a package author to associate a specific XML node set⁶⁾ in the same manifest document that contains it or an XML node set in a different manifest document instance with the parent object containing an pointer instance.</p> <p>A referenced node set must be a valid child of the referencing parent element, both as to kind and multiplicity in a referencing parent's context.</p> <p>This element represents a binding of the kinds of objects defined as children of Manifest.[ManifestMetadata, Organizations, Resources, Manifest, Extension] (see subclause 6.9 of ISO/IEC 12785-1:2009).</p> |
|----------------------|--|

XML instance representation:

```

<...
identifier=" xs:ID [1]"
xlink:type="[0..1]"
xlink:href="[1]"
Allow any attributes from a namespace other than this schema's namespace (strict validation).
>
    Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="IPointer.Type">
  <xs:sequence>
    <xs:group ref=" grpStrict.any " />
  </xs:sequence>
  <xs:attribute name="identifier" type=" xs:ID " use="required"/>
  <xs:attribute ref=" xlink:type " use="optional" default="simple"/>
  <xs:attribute ref=" xlink:href " use="required"/>
  <xs:anyAttribute namespace="##other" processContents="strict"/>
</xs:complexType>

```

6) Node set is one of basic types in W3C's XPath specification (<http://www.w3.org/TR/xpath>).

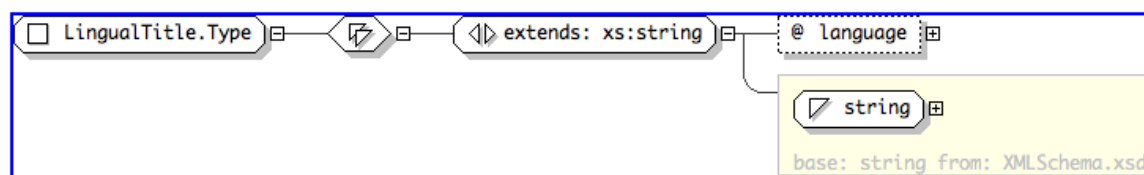
5.2.3.2 Complex Type: **LingualTitle.Type**

| | |
|--------------|---|
| Super-types: | <u>xs:string</u> < LingualTitle.Type (by extension) |
| Sub-types: | None |

| | |
|--|-----------------------------|
| Name | LingualTitle.Type |
| Used by (from the same schema document) | Element <u>lingualTitle</u> |
| Abstract | no |

XML instance representation:

```
<...
language=" xs:string [1]">
xs:string
</...>
```

Diagram:**Schema component representation:**

```
<xs:complexType name="LingualTitle.Type">
  <xs:simpleContent>
    <xs:extension base=" xs:string ">
      <xs:attribute name="language" type=" xs:string " use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

5.2.3.3 Complex Type: **Metadata.Type**

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|---------------------------|
| Name | Metadata.Type |
| Used by (from the same schema document) | Complex Type Variant.Type |
| Abstract | no |

| | |
|----------------------|---|
| Documentation | <p>An instance of the metadata element contains data structures that declare descriptive information about a metadata element's parent only.</p> <p>One or more different metadata models may be declared as child extensions of a metadata element.</p> <p>This element represents a binding of the kinds of child objects defined for Metadata.[Extension] (see subclause 6.7 of ISO/IEC 12785-1:2009).</p> |
|----------------------|---|

XML instance representation:

```
<...>
    Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]
</...>
```

Diagram:**Schema component representation:**

```
<xs:complexType name="Metadata.Type">
  <xs:sequence>
    <xs:group ref=" grpStrict.any " />
  </xs:sequence>
</xs:complexType>
```

5.2.3.4 Complex Type: Variant.Type

| | |
|--------------|------|
| Super-types: | None |
| Sub-types: | None |

| | |
|--|--|
| Name | Variant.Type |
| Used by (from the same schema document) | Element variant |
| Abstract | no |
| Documentation | <p>A variant element is closely analogous to a resource element in ISO/IEC 12785-1. Variant is a container for a alternative resource. A resource may contain references to files that are all of the same type or different types (i.e., file formats).</p> <p>The Variant class points to the alternative resource. Metadata is used to describe the nature of a collection of alternative files and their intended use. Examples include but are not limited to the use of lingual variants, visual or auditory variants, remediation variants, or platform delivery variants.</p> <p>The types of referenced files are specific to a Variant object. Their use is in the context of the parent object containing a variant instance, typically a bound instance of a Resource object from the IMS Content Packaging namespace.</p> |

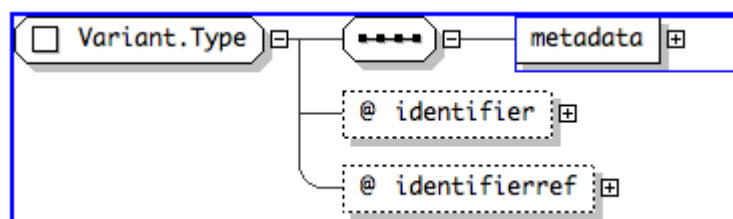
| | |
|--|---|
| | This element represents a binding of the kinds of child objects defined for Resource.[Metadata, File, Dependency, Extension] (see subclause 6.8 of ISO/IEC 12785-1:2009). |
|--|---|

XML instance representation:

```

<...
identifier=" xs:ID [1]"
identifierref=" xs:IDREF [1]">
<metadata> Metadata.Type </metadata> [1]
</...>

```

Diagram:**Schema component representation:**

```

<xs:complexType name="Variant.Type">
  <xs:sequence>
    <xs:element name="metadata" type=" Metadata.Type " minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="identifier" type=" xs:ID " use="required"/>
  <xs:attribute name="identifierref" type=" xs:IDREF " use="required"/>
</xs:complexType>

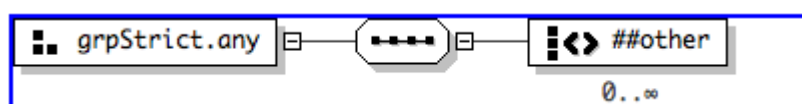
```

5.2.3.5 Model Group: grpStrict.any

| | |
|--|--|
| Name | grpStrict.any |
| Used by (from the same schema document) | Complex Type IPointer.Type , Complex Type, Metadata.Type |
| Documentation | Any namespaced element from any namespace may be included within an "any" element. The namespace for any element that is defined in another standard or specification must be declared in the instance, and the schema must be referenced. The extension has a definition of "strict" i.e. it must have its own namespace. |

XML instance representation:

Allow any elements from a namespace other than this schema's namespace (strict validation). [0..*]

Diagram:

Schema component representation:

```

<xs:group name="grpStrict.any">
  <xs:sequence>
    <xs:any namespace="##other" processContents="strict" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

```

6 Using schematron

To help implementors enforce information model constraints that the XSDs cannot enforce, a Schematron schema has been made available as part of the specification's documentation set.

The two XSDs (core and extension) are not linked i.e. one does not import the other. This is because linking creates an ambiguous binding due to support for extensibility. The reason for the ambiguity is that the need for both XSDs to allow extensibility could lead to one XSD allowing the import of structures from the other namespace that the other XSD forbids. This lack of formal linkage of one XSD with another means that the constraints in ISO/IEC 12785-1 regarding components of the extension is not enforced in the binding. The only constraint that can be enforced by the XSDs is that the components of the new extension can ONLY be inserted at an extensibility point within the core content packaging binding.

The schematron schema only encodes the constraints on where place in a manifest document where elements from the extended content packaging namespace are allowed to occur. The schematron schema will also provide some feedback about violations of those constraints to schematron compliant applications.

Note: It should be noted that ISO/IEC 12785 Content Packaging compliant systems are not required to support Schematron; the Schematron schema is supplied as an extra implementation aid. It also worth noting that Schematron is trivially implementable on top of an existing XSLT parser. Finally, because there is no general method of associating Schematron schemas with XML instances, the task of applying the Schematron schema to an instance is not specified here.

Annex A
(informative)**Document Provenance**

| | |
|-----------------------------|---|
| Title | IMS Content Packaging XML Binding |
| Editors | Colin Smythe (IMS), Boyd Nielsen (Independent) |
| Team Co-Leads | Jan Poston Day (Blackboard), Wilbert Kraan (JISC/CETIS), Nigel Ward (DEST) |
| Version | v1.2 (CM/DN Draft v2.0) |
| Version Date | 01 March 2007 |
| Status | CM/DN Draft |
| Summary | This document describes the XML Binding of the Content Packaging specification. |
| Revision Information | 31 October 2006. This version supersedes Public Draft v1.0 released in November 2005. |
| Purpose | This document has been reviewed by the IMS Technical Board and is made available to IMS Contributing Members and IMS Developer Network Subscribers for review and feedback. |
| Document Location | Content Packaging Document Library. |

Annex B (informative)

Listing of the XSDs

This annex provides links to XSD documents and indicates the associated namespaces.

B.1 Listing of the Content Package v1.2 XSD

The core content packaging XSD is contained in the file: imscp_v1p2.xsd. The default namespace is 'http://www.imsglobal.org/xsd/imscp_v1p2' and the default schema location is 'http://www.imsglobal.org/xsd/imscp_v1p2.xsd'.

B.2 Listing of the Content Package Extensions v1.2 XSD

The content packaging extension XSD is contained in the file: imscp_extensionv1p2.xsd. The default namespace is 'http://www.imsglobal.org/xsd/imscp_extensionv1p2' and the default schema location is 'http://www.imsglobal.org/xsd/imscp_extensionv1p2.xsd'.

Bibliography

- [1] 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
- [2] 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
- [3] 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
- [4] IMS Binding Auto-generation Toolkit Manual v1.0, C. Smythe, IMS Global Learning Consortium, July 2006.
- [5] IMS Vocabulary Definition Exchange Information Model Specification Final Release v1.0, A. Cooper, IMS Global Learning Consortium, February 2004
- [6] IMS Vocabulary Definition Exchange XML Binding Specification Final Release v1.0, A. Cooper, IMS Global Learning Consortium, February 2004
- [7] W3C RDF/XML Syntax Specification (Revised) (2004)
- [8] W3C XML Schema Part 2: Datatypes Second Edition (2004)
- [9] W3C Recommendation, Namespaces in XML 1.0 (Second Edition), 16 August 2006
- [10] ISO 639-2:1998, *Codes for the representation of names of languages — Part 2: Alpha-3 code*
- [11] ISO 3166-1:1997, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*
- [12] ISO/IEC 10646:2003, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*
- [13] ISO/IEC 19757-3:2006, *Information technology — Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation — Schematron*

