



IMS Learning Design XML Binding

Version 1.0 Final Specification

IPR and Distribution Notices

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: http://www.imsglobal.org/ipr/imsipr_policyFinal.pdf.

Copyright © 2003 IMS Global Learning Consortium. All Rights Reserved.

If you wish to copy or distribute this document, you must complete a valid Registered User license registration with IMS and receive an email from IMS granting the license to distribute the specification. To register, follow the instructions on the IMS website: <http://www.imsglobal.org/specificationdownload.cfm>.

This document may be copied and furnished to others by Registered Users who have registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS project group.

Use of this specification to develop products or services is governed by the license with IMS found on the IMS website: <http://www.imsglobal.org/license.html>.

The limited permissions granted above are perpetual and will not be revoked by IMS or its successors or assigns.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

Copyright © 2003 by IMS Global Learning Consortium, Inc.
All Rights Reserved.

The IMS Logo is a trademark of IMS Global Learning Consortium, Inc.

Document Name: IMS Learning Design XML Binding

Date: 20 January 2003

Table of Contents

1.	INTRODUCTION	6
1.1	SCOPE AND CONTENT	6
1.2	STRUCTURE OF THIS DOCUMENT	6
1.3	NOMENCLATURE	6
2.	XML BASICS	7
2.1	ELEMENTS	7
2.1.1	<i>Element Contents</i>	7
2.1.2	<i>Element Attributes</i>	7
2.1.3	<i>Element Names</i>	7
2.2	DOCUMENT TYPE DEFINITIONS	8
2.3	XML SCHEMAS	8
2.4	VALID CHARACTER SETS	8
2.5	SPECIAL HANDLING REQUIREMENTS	8
2.5.1	<i>XML Reserved Characters</i>	8
2.5.2	<i>White Space Handling</i>	9
3.	DESCRIPTION OF THE XML SCHEMA BINDING	10
4.	NARRATIVE DESCRIPTION OF LEARNING DESIGN SCHEMA LEVEL A	11
4.1	<LEARNING-DESIGN> ELEMENTS	11
4.1.1	<title>	12
4.1.2	<learning-objectives>	12
4.1.3	<prerequisites>	12
4.1.4	<components>	13
4.1.5	<method>	14
4.1.6	<metadata>	14
4.2	<COMPONENTS> ELEMENTS	15
4.2.1	<roles>	15
4.2.2	<activities>	15
4.2.3	<environments>	16
4.3	<ROLES> ELEMENTS	17
4.3.1	<learner>	17
4.3.2	<staff>	18
4.4	<ACTIVITIES> ELEMENTS	20
4.4.1	<learning-activity>	20
4.4.2	<support-activity>	20
4.4.3	<activity-structure>	21
4.5	<LEARNING-ACTIVITY> ELEMENTS	22
4.5.1	<title>	22
4.5.2	<learning-objectives>	23
4.5.3	<prerequisites>	23
4.5.4	<environment-ref>	23
4.5.5	<activity-description>	24
4.5.6	<complete-activity>	24
4.5.7	<on-completion>	25
4.5.8	<metadata>	25
4.6	<SUPPORT-ACTIVITY> ELEMENTS	26
4.6.1	<title>	26

4.6.2	<role-ref>	26
4.6.3	<environment-ref>.....	26
4.6.4	<activity-description>	26
4.6.5	<complete-activity>	27
4.6.6	<on-completion>	27
4.6.7	<metadata>.....	27
4.7	<ACTIVITY-STRUCTURE> ELEMENTS	27
4.7.1	<title>	27
4.7.2	<information>.....	27
4.7.3	<environment-ref>.....	28
4.7.4	<learning-activity-ref>	28
4.7.5	<support-activity-ref>	28
4.7.6	<unit-of-learning-href>	28
4.7.7	<activity-structure-ref>	28
4.7.8	<metadata>.....	28
4.8	<ENVIRONMENTS> ELEMENTS	29
4.8.1	<environment>	29
4.9	<SERVICE> ELEMENTS	32
4.9.1	<send-mail>.....	32
4.9.2	<conference>	33
4.9.3	<index-search>.....	36
4.9.4	###other	37
4.10	<METHOD> ELEMENTS	37
4.10.1	<play>.....	38
4.10.2	<complete-unit-of-learning>	38
4.10.3	<on-completion>	39
4.11	<PLAY> ELEMENTS	40
4.11.1	<title>	40
4.11.2	<act>.....	40
4.11.3	<complete-play>.....	41
4.11.4	<on-completion>	41
4.11.5	<metadata>.....	41
4.12	<ACT> ELEMENTS	42
4.12.1	<title>	42
4.12.2	<role-part>	42
4.12.3	<complete-act>	43
4.12.4	<on-completion>	44
4.12.5	<metadata>.....	44
5.	NARRATIVE DESCRIPTION OF LEARNING DESIGN SCHEMA LEVEL B	45
5.1	<PROPERTIES> ELEMENTS	45
5.1.1	<loc-property>	46
5.1.2	<locpers-property>.....	47
5.1.3	<locrole-property>	48
5.1.4	<globpers-property>	49
5.1.5	<glob-property>	50
5.1.6	<property-group>	51
5.2	<WHEN-PROPERTY-VALUE-IS-SET> ELEMENTS	51
5.2.1	<property-ref>	52

	5.2.2	<i><property-value></i>	52
5.3		<CHANGE-PROPERTY-VALUE> ELEMENTS	53
	5.3.1	<i><property-ref></i>	53
	5.3.2	<i><property-value></i>	53
5.4		<MONITOR> ELEMENTS	54
	5.4.1	<i><role-ref></i>	54
	5.4.2	<i><self></i>	54
	5.4.3	<i><title></i>	54
	5.4.4	<i><item></i>	54
	5.4.5	<i><metadata></i>	55
5.5		<EMAIL-DATA> ADDITIONAL ATTRIBUTES	55
5.6		<CONDITION> ELEMENTS	55
	5.6.1	<i><title></i>	56
	5.6.2	<i><if></i>	57
	5.6.3	<i><then></i>	57
	5.6.4	<i><else></i>	57
	5.6.5	<i><metadata></i>	57
5.7		{EXPRESSION} SCHEMA GROUP	58
	5.7.1	<i><is-member-of-role></i>	60
	5.7.2	<i><is></i>	60
	5.7.3	<i><is-not></i>	60
	5.7.4	<i><and></i>	60
	5.7.5	<i><or></i>	61
	5.7.6	<i><sum></i>	61
	5.7.7	<i><subtract></i>	61
	5.7.8	<i><multiply></i>	61
	5.7.9	<i><divide></i>	61
	5.7.10	<i><greater-than></i>	62
	5.7.11	<i><less-than></i>	62
	5.7.12	<i><users-in-role></i>	62
	5.7.13	<i><no-value></i>	62
	5.7.14	<i><time-unit-of-learning-started></i>	62
	5.7.15	<i><daytime-activity-started></i>	62
	5.7.16	<i><current-datetime></i>	63
	5.7.17	<i><complete></i>	63
	5.7.18	<i><not></i>	64
5.8		{THENMODEL} SCHEMA GROUP	65
	5.8.1	<i><show></i>	65
	5.8.2	<i><hide></i>	67
	5.8.3	<i><change-property-value></i>	68
5.9		<CALCULATE>	68
5.10		<WHEN-CONDITION-TRUE>	69
	5.10.1	<i><role-ref></i>	69
	5.10.2	<i><expression></i>	69
5.11		<GLOBAL-ELEMENTS> ELEMENTS	69
	5.11.1	<i><view-property></i>	70
	5.11.2	<i><view-property-group></i>	70
	5.11.3	<i><set-property></i>	70

5.11.4	<set-property-group>	71
5.12	GLOBAL ATTRIBUTE 'CLASS'	72
5.13	DATA TYPES.....	72
5.14	RESTRICTION TYPES.....	73
6.	NARRATIVE DESCRIPTION OF LEARNING DESIGN SCHEMA LEVEL C.....	74
6.1	<NOTIFICATION> ELEMENTS	74
6.1.1	<email-data>	75
6.1.2	<learning-activity-ref>	75
6.1.3	<support-activity-ref>	75
6.1.4	<subject>	75
6.2	<NOTIFICATION> EXTENSIONS	75
6.2.1	<on-completion>	75
6.2.2	<then>.....	76
6.2.3	<global-elements>	76
7.	NORMATIVE POINTS	77
7.1	XML SCHEMA FILE NAMING CONVENTION	77
7.2	NAMESPACE URI (NAMESPACE IDENTIFIER).....	77
7.2.1	<i>Location of the Current Learning Design Binding Instance.....</i>	<i>77</i>
7.2.2	<i>Location of Versioned Learning Design Binding Instances.....</i>	<i>77</i>
7.3	NAMESPACE NAME.....	77
7.4	VERSIONING A BINDING INSTANCE.....	78
7.5	LOCALIZATION	78
7.5.1	<i>Localization Binding Structure Name and Values</i>	<i>78</i>
7.5.2	<i>Localizing this Document.....</i>	<i>78</i>
7.6	EXTENSIBILITY	79
	ABOUT THIS DOCUMENT	80
	LIST OF CONTRIBUTORS	80
	REVISION HISTORY	81
	INDEX.....	82

1. Introduction

The IMS Learning Design XML Binding guide is a collection of notes and support information that describes how the IMS Learning Design Information Model v1.0 is represented as a set of XML elements. The information model describes these elements in an abstract manner, while this document explains how they are bound using XML Schema conventions.

1.1 Scope and Content

This document is the IMS Learning Design XML Binding Specification. As such it will be used as the basis for the production of the following documents:

- IMS Learning Design XML Schema.
- IMS Learning Design Best Practice Guide.

This binding has been derived from the agreed IMS Learning Design Information Model and conforms to the XML 1.0 Specification of the W3C.

1.2 Structure of this Document

The structure of this document is:

2. XML Basics	A brief description of the components within an XML schema.
3. Normative Description of the XML Binding	A detailed description of the realization of the Learning Design Information Model in XML.
4. Narrative Description of Learning Design Schema Level A	The description of the elements and attributes used within the Level A XML binding.
5. Narrative Description of Learning Design Schema Level B	The description of the elements and attributes used within the Level B XML binding.
6. Narrative Description of Learning Design Schema Level C	The description of the elements and attributes used within the Level C XML binding.
7. Normative Points	Important normative statements that must be maintained by implementations of this XML binding.

1.3 Nomenclature

CDATA	Character Data
DTD	Document Type Definition
PCDATA	Parsed Character Data
W3C	World Wide Web Consortium
XML	Extensible Mark-up Language

2. XML Basics

The Learning Design concepts defined in the Information Model are intended to be represented using a hierarchy. Hierarchical models are convenient for representing data consisting of many elements and sub-elements. XML (eXtensible Markup Language) is a widely used method for representing hierarchical models and has been adopted by IMS as the basis for interoperability. An XML document is a hierarchy of *elements* which may have *contents* and *attributes*.

2.1 Elements

An element is a structured component of a document identified by a *tag name*. When a tag name is shown as “<TAGNAME>”, with angle brackets before and after the tag name, it serves as the *start-tag* to mark the beginning of an element. When that same tag name has a forward slash “/” added, it serves as an *end-tag* such as “</TAGNAME>”. An element may have contents between its start and end-tags, and may have one or more *attributes*. When an XML element has a start and end-tag (also called an *opening* and *closing* tag) with a common name, it is considered to be “well-formed” XML. The contents of an element are placed between the start and end-tags as shown here:

```
<TAGNAME>contents</TAGNAME>
```

2.1.1 Element Contents

An element may contain other elements, Parsed Character Data (PCDATA), Character Data (CDATA), or a mixture of PCDATA and elements. The allowable contents of an element are its *content model*. XML parsers treat PCDATA with their special or reserved meaning unless they are specifically marked (or “escaped”). In contrast, CDATA can use special or reserved characters without having to escape them, as CDATA is not read by XML parsers.

2.1.2 Element Attributes

An attribute provides additional information about an element. Attributes are a way of attaching characteristics or properties to the elements of a document. An element may have more than one attribute. Attributes are contained within the start tag of an element. Attributes are represented by an attribute name followed by an equal sign and the attribute value in quotation marks:

```
<timeframe>  
  <begin restrict="1"> 1999-07-23 </begin>  
</timeframe>
```

In this example, the timeframe element contains another element: the opening tag. The opening tag has one attribute “restrict”, with the value “1”. The value for the element BEGIN is “1999-07-23”. These two elements then make up a timeframe begin date. Attributes may appear in any order within the opening tag.

2.1.3 Element Names

Each element has a unique name, referred to as the tag name. XML is case-sensitive in its processing of tag names. The IMS Learning Design XML Binding Specification adheres to the following tag name rules:

- All tag names will conform to the rules for element naming as given within the XML 1.0 Specification.
- Names beginning with “XML” in any case or mix of cases are not permitted.
- The IMS binding will use lower case leading camelCase tag and element names.
- Element names may not include words reserved by the XML specification. These include:
 - DOCTYPE
 - ELEMENT
 - ATTLIST
 - ENTITY
- Tag names defined by this IMS Binding may not be redefined.

2.2 Document Type Definitions

The tag name, content model, and attributes of elements were historically defined in a *Document Type Definition* (DTD) statement. A DTD is a kind of schema. It may exist as an external file or a block of text internal to an XML document. However, the DTD schema was developed before object-oriented programming concepts became prevalent within software development communities and before the use of entities drawn from multiple namespaces became a requirement. Those communities began working on other schema representations to provide more object-like structures and procedures and to support the flexible use of multiple namespaces than could be realized within the constraints of the Document Type Definition schema for describing and structuring the contents of XML documents. One such schema language has become today's preferred representation language: XML Schema produced by the Worldwide Web Consortium (W3C).

2.3 XML Schemas

A schema is a formal specification of element names that indicates which elements are allowed in an XML instance, and in which combinations. New schema languages, such as those defined in the XML-Schemas Working Group, provide the same baseline functionality as a DTD. However, because these schema languages are extensible, developers can augment them with additional information, such as data types, inheritance, and presentation rules. This makes schema languages far more powerful than DTDs. For more information about XML schemas, there is a link to the W3C XML Schema Recommendation in the Appendix.

This specification defines a W3C XML Schema (imsld_bindv1p0.xsd) as a non-normative reference. Some XML editors may make use of these schemas to help guide the developer in creating the proper elements at the proper locations in an XML file. Other developers will make use of the schemas to validate their XML instances and/or to define extensions to the IMS Learning Design Binding. Details of the construction of schemas are outside the scope of this document.

2.4 Valid Character Sets

An IMS Learning Design definition instance must use UTF-8 encoding of the character sets as defined in ISO 10646. See the XML Version 1.0 for more details on the specification of well-formed XML.

2.5 Special Handling Requirements

2.5.1 XML Reserved Characters

Some characters used in XML must be *escaped* when used outside of their XML-defined usage as found in section 2.4 of the XML 1.0 Specification. These characters are ampersand (&), less than (<), greater than (>), apostrophe ('), and the double-quote character (").

These characters may be represented using either numeric character references or the strings "&", "<", ">", "'," and """.

Below is a more complete quote from the W3C XML standards:

Quote from Extensible Markup Language (XML) 1.0
W3C Recommendation 10-February-1998
2.4 Character Data and Markup

"Text consists of intermingled character data and markup. Markup takes the form of start-tags, end-tags, empty-element tags, entity references, character references, comments, CDATA section delimiters, document type declarations, and processing instructions.

"All text that is not markup constitutes the character data of the document.

“The ampersand character (&) and the left angle bracket (<) may appear in their literal form only when used as markup delimiters, or within a comment, a processing instruction, or a CDATA section. They are also legal within the literal entity value of an internal entity declaration; see "4.3.2 Well-Formed Parsed Entities". If they are needed elsewhere, they must be escaped using either numeric character references or the strings "&" and "<" respectively. The right angle bracket (>) may be represented using the string ">", and must, for compatibility, be escaped using ">" or a character reference when it appears in the string "]]>" in content, when that string is not marking the end of a CDATA section.

“In the content of elements, character data is any string of characters, which does not contain the start-delimiter of any markup. In a CDATA section, character data is any string of characters not including the CDATA-section-close delimiter, "]]>".

“To allow attribute values to contain both single and double quotes, the apostrophe or single-quote character (') may be represented as "'"; and the double-quote character (") as """.

2.5.2 White Space Handling

Questions often arise as to whether web-based data transmission tools might inadvertently strip off or transform some of the white space characters embedded in data transmitted between systems using XML. To eliminate concern about this issue, refer to the following quote from the W3C XML standards, which indicate that all white space must be preserved where it is part of the data.

Quote from Extensible Markup Language (XML) 1.0

W3C Recommendation 10-February-1998

2.10 White Space Handling

“In editing XML documents, it is often convenient to use "white space" (spaces, tabs, and blank lines, denoted by the non-terminal S in this specification) to set apart the mark-up for greater readability. Such white space is typically not intended for inclusion in the delivered version of the document. On the other hand, "significant" white space that should be preserved in the delivered version is common, for example in poetry and source code.

“An XML processor must always pass all characters in a document that are not mark-up through to the application. A validating XML processor must also inform the application which of these characters constitute white space appearing in element content.

“A special attribute named xml:space may be attached to an element to signal an intention that in that element, white space should be preserved by applications. In valid documents, this attribute, like any other, must be declared if it is used. When declared, it must be given as an enumerated type whose only possible values are "default" and "preserve". For example:

“<!ATTLIST poem xml:space (default|preserve) 'preserve'>

“The value "default" signals that applications' default white-space processing modes are acceptable for this element; the value "preserve" indicates the intent that applications preserve all the white space. This declared intent is considered to apply to all elements within the content of the element where it is specified, unless overridden with another instance of the xml:space attribute. ”

3. Description of the XML Schema Binding

This section of the specification uses a series of graph segments, accompanied by simple narrative, to describe the XML format of the XSD binding for the Learning Design Definition Model. XSD documents that implement this *abstract* format are referenced as non-normative parts of this specification.

These data elements and their relationships from the information model are expressed in XML Schema by means structuring entities called elements, attributes, and groups. These structuring entities are further typified as complex types or simple types. The XML Schema structuring entities comprise a binding of an information model when expressed in a special instance of an XML document – an XML Schema definition (XSD).

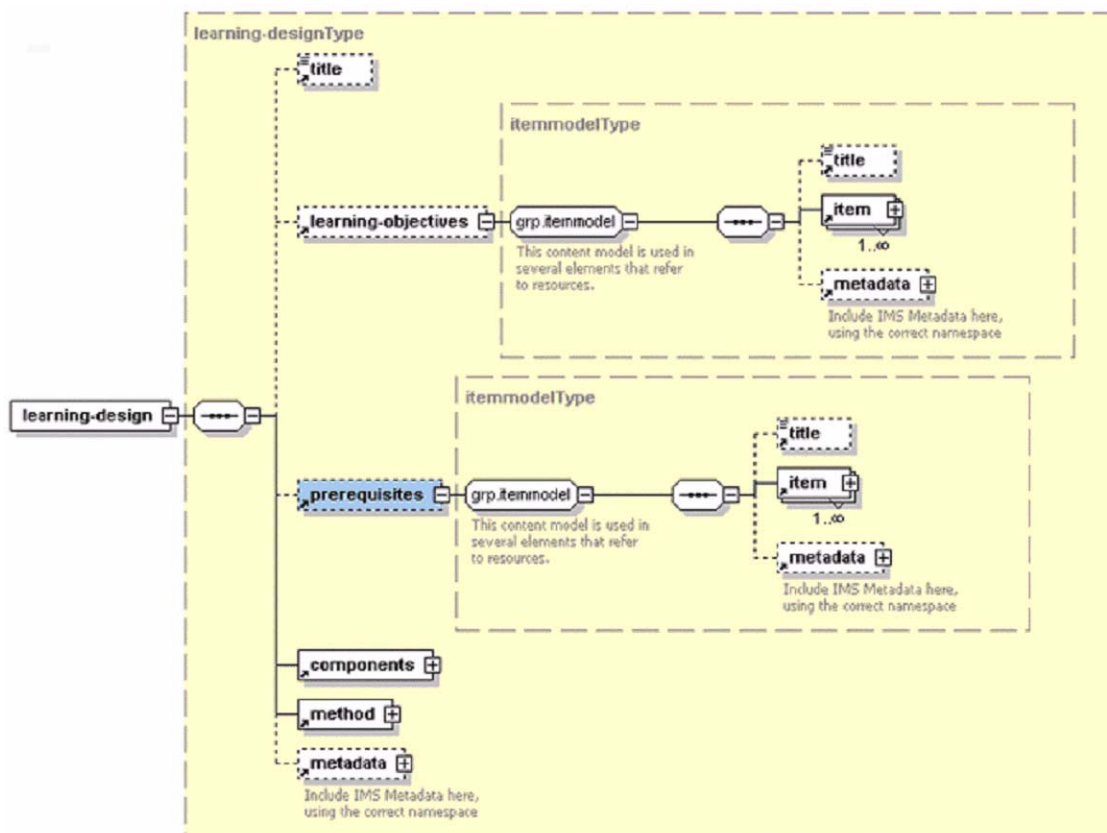
The Learning Design Definition Model also defines values or vocabulary terms required by certain data model elements. These controlled lists of terms or values can be expressed in XML Schema as restrictions placed on the values that may be declared for a given structuring element.

It is often necessary for bindings to create structuring entities with names that are not present in an information or data model that is being expressed in XML Schema. These special structuring enable the grouping and use of elements as defined in a data model.

These graphs will show individual elements encapsulated within larger structures, like complex types. Structures without child elements are not represented separately. The reader is referred to a binding document instance (.xsd) for the full lexical representation of all structures and values.

4. Narrative Description of Learning Design Schema Level A

4.1 <learning-design> Elements



Description: This element specifies the learning design.

Multiplicity: Undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **version (optional).** A version number. Data type = string.
- **uri (mandatory).** Specifies a URI. Data type = string.
- **level (optional).** Specifies the lowest level of Learning Design that the document instance is valid. The letter is specified with one of the following characters: A, B, C, a, b, or c. Data type = string.
- **sequence-used (mandatory).** Boolean, when set to 'true' IMS Simple Sequencing is included at the appropriate places in the document instance. Data type = token group.

Elements:

- title
- learning-objectives
- prerequisites
- components
- method

- metadata

4.1.1 <title>

Description: See IMS CP. A short name given to the resource, suitable for rendering in user-agents.

Multiplicity: The <title> element is optional and occurs zero or 1 time.

Attributes:

Example:

```
<learning-design identifier="LD_boeing_simplified" uri="URI" level="A"
xmlns="http://www.imsglobal.org/xsd/imsld_v1_p0">
  <title>Boeing Fuel Valve Removal simplified</title>
</learning-design>
```

4.1.2 <learning-objectives>

Description: Learning objectives describe the intended outcome for learners. Learning-objectives & prerequisites contain a standard organization of items, referring to resources or sub manifests. Resource type connected to learning objectives and prerequisites can be webcontent, imslcontent or it can point to an IMS Reusable Definition of Competency or Educational Objective (RDCEO) Schema. There are two locations where learning-objectives & prerequisites are specified: - At the level of the learning design (in the root of learning-design) - At the level of learning-activities (within learning-activities). The first ones are a more general description; the second ones are more concrete. There are two types of learning-objectives: -1- human readable descriptions (the items point to text resources) -2- machine-readable specifications. These are addressed through the href attribute of the resources pointed at. The learning-objectives schema's could be user-defined or fixed by an organization. In the latter case, the texts of the learning objectives are referred to (through href).

Multiplicity: The <learning-objectives> element is optional and occurs zero or 1 time.

Attributes:

- {itemmodel} (mandatory). See IMS CP. A Schema group. Data type = group.

Elements:

- title
- item
- metadata

Example:

```
<learning-objectives>
  <item isvisible="true" parameters="" identifierref="" identifier="LOB_learning_objectives"
xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1"></item>
</learning-objectives>
```

4.1.3 <prerequisites>

Description: Prerequisites are the entry-requirements for students, such as any pre-knowledge needed. For the item formats see the description of the element <learning-objectives>.

Multiplicity: The <prerequisites> element is optional and occurs zero or 1 time.

Attributes:

- {itemmodel} (mandatory). See IMS CP. A Schema group. Data type = group.

Elements:

- title
- item
- metadata

Example:

```
<prerequisites>
  <item isvisible="true" parameters="" identifierref="" identifier="PREQ_prerequisites"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"></item>
</prerequisites>
```

4.1.4 <components>

Description: Specifies the building blocks used in the method section.

Multiplicity: The <components> element is mandatory and occurs only 1 time.

Attributes: None

Elements:

- roles
- activities
- environments

Example:

```
<components>
  <roles>
    <learner identifier="R_learner"/>
  </roles>
  <activities>
    <learning-activity isvisible="true" identifier="LA_fuel_valve_lesson_intro">
      <activity-description>
<item isvisible="true" parameters="" identifierref="" identifier="I_fuel_valve_lesson_intro"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
      </activity-description>
    </learning-activity>

    <learning-activity isvisible="true" identifier="LA_fuel_valve_theory">
      <activity-description>
        <item isvisible="true" parameters="" identifierref=""
identifier="I_fuel_valve_theory" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
        </activity-description>
      </learning-activity>

    <activity-structure identifier="AS_introduction" number-to-select="2"
structure-type="sequence">
      <title e-dtype="string" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
      <learning-activity-ref ref="LA_fuel_valve_lesson_intro"/>
      <learning-activity-ref ref="LA_fuel_valve_theory"/>
    </activity-structure>

    <activity-structure identifier="AS_fuel_valve_lessons" number-to-select="2"
structure-type="selection">
      <title e-dtype="string" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
      <learning-activity-ref ref="LA_lesson_hazards"/>
      <learning-activity-ref ref="LA_lesson_components"/>
    </activity-structure>
  </activities>
```

```

    <environments>
      <environment identifier="E_interactive_electronic_training_manual">
<title e-dtype="string" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1">Interactive Electronic
Technical Manual</title>
      </environment>
    </environments>
  </components>

```

4.1.5 <method>

Description: The <method> element contains a sequence of elements for the definition of the dynamics of the learning process. It consists of a ‘play’ (which could be interpreted as the runscript for the unit of learning) and a ‘statement’ for the completion of the unit of learning.

Multiplicity: The <method> element is mandatory and occurs only 1 time.

Elements:

- play
- complete-unit-of-learning
- on-completion

Example:

```

<method>
  <play identifier="P_Boeing_simplified" isvisible="true">
    <act identifier="ACT_individualized_learning">
      <role-part identifier="RP_individualized_learning">
        <role-ref ref="R_learner"/>
        <activity-structure-ref ref="AS_boeing_simplified" />
      </role-part>
      <complete-act>
        <when-role-part-completed ref="RP_individualized_learning"/>
      </complete-act>
    </act>
    <complete-play>
      <when-last-act-completed/>
    </complete-play>
  </play>
</method>

```

4.1.6 <metadata>

Description: Include IMS Meta-Data here, using its namespace.

Multiplicity: The <metadata> element is optional and occurs zero, once, or more times.

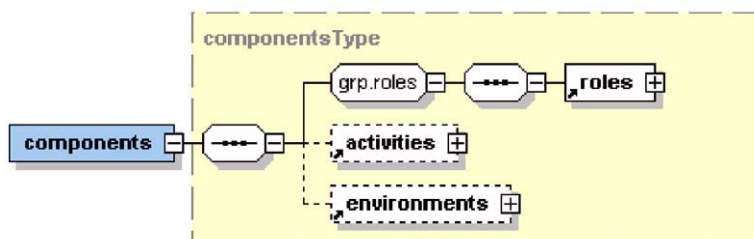
Example:

```

<metadata>
  <schema>IMS Metadata</schema>
  <schemaversion>1.2</schemaversion>
  <imsmd:general>
    <imsmd:title>The Versailles Experience</imsmd:title>
  </imsmd:general>
</metadata>

```

4.2 <components> Elements



4.2.1 <roles>

Description: The <roles> element specifies the roles distinguished in the learning design. Roles contains a sequence to declare the two general roles: learner & staff. A href can be provided when referring to a global role (e.g. a role defined by an institute). This is obligatory when specifying a global role and connected globrole-properties. Global roles are specified with the href attribute. The rest of the declaration, like information, is local. It is not possible to declare global roles in a learning design. This is just an organizational issue and is nothing more or less than providing absolute URIs for roles. The URI does not necessarily point to a resource on the location of the address, its just used as a worldwide unique identifier. The attribute 'identifier' on roles can be used to refer to the whole group of all roles within the learning-design (learners and staff). In every learning-design at least one learner role is specified. In institutional installations the role names are fixed. For instance in most universities, the role-identifier for learners is: 'student'.

Multiplicity: The <roles> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID).
- **href (optional).** An absolute URI to identify the role.
- **min-persons (optional).** A non-negative number of persons.
- **max-persons (optional).** A non-negative number of persons.

Elements:

- learner
- staff

Example:

```

< roles>
  <learner identifier="Public_Adminstration_Student">
    <title>Public Adminstration Student</title>
  </learner>
  <staff identifier="Tutor">
    <title>Tutor</title>
  </staff>
  <staff identifier="Examiner">
    < title>Examiner</title>
  </staff>
</roles>

```

4.2.2 <activities>

Description: The <activities> element contains a choice for different activity definitions such as learning activities and support activities. It also provides for the definition of activities structures..

Multiplicity: The <activities> element is optional and occurs zero or 1 time.

Elements:

- learning-activity
- support-activity
- activity-structure

Example:

```
<activities>
  <learning-activity identifier="Versailles_Overview">
    <title>Versailles Experience Aims and Objectives</title>
    <environment-ref ref="Versailles_Aims"/>
    <activity-description>
      <item identifier="Versailles_Overview_AD_res"/>
    </activity-description>
  </learning-activity>
</activities>
```

4.2.3 <environments>

Description: The <environments> element is a container for the <environment> elements.

Multiplicity: The <environments> element is optional and occurs zero or 1 time.

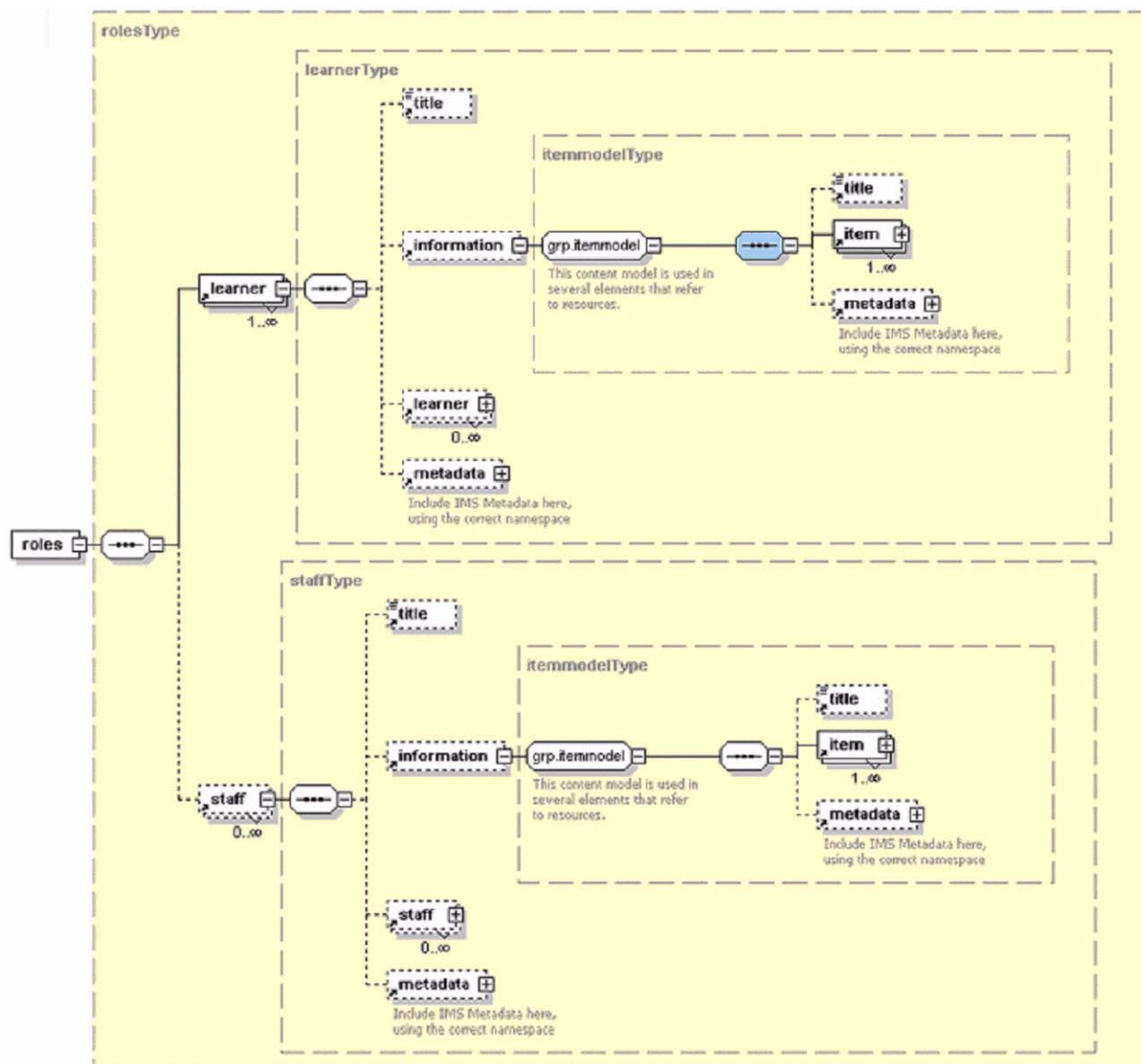
Elements:

- environment

Example:

```
<imsld:environments>
  <imsld:environment identifier="Versailles_Aims">
    <imsld:title>Aims of the Versailles Experience</imsld:title>
    <imsld:learning-object identifier="Versailles_Aims_LO">
      <imsld:item identifier="V_Aims_item" identifierref="V_Aims_res"/>
    </imsld:learning-object>
  </imsld:environment>
</imsld:environments>
```


4.3 <roles> Elements



4.3.1 <learner>

Description: In every learning design there is at least one learner-role. Learners can be ‘nested’, meaning that a role may be divided in sub roles. The title in the learner model is used to provide the name for the role. For example, in an educational game you can distinguish roles like chair and participant as sub roles of student.

Multiplicity: The <learner> element is mandatory and occurs 1 or more times.

Attributes:

- **create-new (mandatory).** This attribute indicates whether multiple occurrences of this role may be created during run-time. When the attribute has the value "not-allowed" then there is always one and only one instance of the role. If the value is "allowed" a mechanism in the run-time system should be provided to create new instances of this role. If a new instance of a role is created, new instances for all available sub-roles of that role are created as well. Data type = token group.
- **href (optional).** Refers to a URI. Data type = string.
- **identifier (mandatory).** An identifier that is unique within the learning design file (ID).

- **match-persons (optional).** This attribute is used when there are several sub roles (e.g. chair, secretary, member). Persons can be matched exclusively to the sub roles, meaning that a person, who has the role of chair, may not be bound to one of the other roles at the same time. When it is not exclusive, persons may be bound to more than one sub role (this is the default situation). Data type = token group.
- **max-persons (optional).** Specifies the maximum number of persons bound to the role before starting the run. When the attribute min-persons and max-persons are empty, there are no restrictions. Data type = string.
- **min-persons (optional).** Specifies the minimum number of persons bound to the role before starting a run. When the attribute min-persons and max-persons are empty, there are no restrictions. Data type = string.

Elements:

- title
- information
- learner
- metadata

Example:

```
<learner identifier="Learner">
  <title>Learner</title>
</learner>
<learner identifier="GB">
  <title>Great Britain</title>
</learner>
<learner identifier="USA">
  <title>U.S.A.</title>
</learner>
```

4.3.1.1 <title>

See <title> above

4.3.1.2 <information>

Description: The <information> element specifies the set of items pointing to the resource(s) where the information can be found.

Multiplicity: The <information> element is optional and occurs zero or 1 time.

Elements:

- title
- item
- metadata

4.3.1.3 <learner>

See <learner> above.

4.3.1.4 <metadata>

See <metadata> above.

4.3.2 <staff>

Description: Staff members can be ‘nested’, meaning that a role may be divided in sub roles. For example, in an educational institute you may distinguish roles like tutors, assessors, mentors, evaluators, etc.

Multiplicity: The <staff> element is mandatory and occurs 1 or more times.

Attributes:

- **create-new (mandatory).** This attribute indicates whether multiple occurrences of this role may be created during run-time. When the attribute has the value "not-allowed" then there is always one and only one instance of the role. If the value is "allowed" a mechanism in the run-time system should be provided to create new instances of this role. If a new instance of a role is created, new instances for all available sub-roles of that role are created as well. Data type = token group.
- **href (optional).** Refers to a URI. Data type = string.
- **identifier (mandatory).** An identifier that is unique within the learning design file (ID).
- **match-persons (optional).** This attribute is used when there are several sub roles (e.g. chair, secretary, member). Persons can be matched exclusively to the sub roles, meaning that a person, who has the role of chair, may not be bound to one of the other roles at the same time. When it is not exclusive, persons may be bound to more than one sub role (this is the default situation). Data type = token group.
- **max-persons (optional).** Specifies the maximum number of persons bound to the role before starting the run. When the attribute min-persons and max-persons are empty, there are no restrictions. Data type = string.
- **min-persons (optional).** Specifies the minimum number of persons bound to the role before starting a run. When the attribute min-persons and max-persons are empty, there are no restrictions. Data type = string.

Elements:

- title
- information
- staff
- metadata

Example:

```
<imsld:staff identifier="Support_Staff">
    <imsld:title>Support Staff</imsld:title>
</imsld:staff>
<imsld:staff identifier="Teacher">
    <imsld:title>Teacher</imsld:title>
</imsld:staff>
<imsld:staff identifier="Recorder">
    <imsld:title>Recorder</imsld:title>
</imsld:staff>
```

4.3.2.1 <title>

See <title> above.

4.3.2.2 <information>

See <information> above.

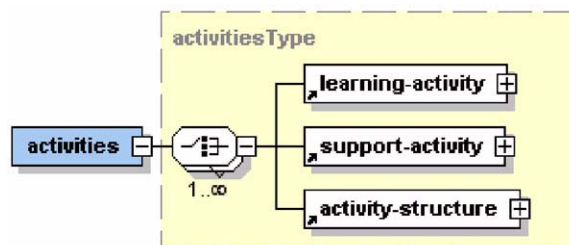
4.3.2.3 <learner>

See <learner> above.

4.3.2.4 <metadata>

See <metadata> above.

4.4 <activities> Elements



4.4.1 <learning-activity>

Description: The <learning-activity> element contains a sequence of elements for learning-activity definitions.

Multiplicity: The <learning-activity> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **isvisible (mandatory).** True or false, initial visibility attribute (see IMS CP). Data type = token group.
- **parameters (optional).** See IMS CP. Data type = string.

Elements:

- title
- learning-objectives
- prerequisites
- environment-ref
- activity-description
- complete-activity
- on-completion
- metadata

Example:

```
<learning-activity identifier="Versailles_Overview">
  <title>Versailles Experience Aims and Objectives</title>
  <environment-ref ref="Versailles_Aims"/>
  <activity-description>
    <item identifier="Versailles_Overview_AD_res"/>
  </activity-description>
</learning-activity>
```

4.4.2 <support-activity>

Description: The <support-activity> element contains a sequence of elements to define support-activities. A support activity can optionally be connected to a role. This means that the activity is repeated for every member in the supported role (learner/staff). Support activities are typically performed by staff members (e.g. tutors) to support learners. In some pedagogical models however learners can support learners (peer to peer teaching). It is also possible that staff members support staff members. When the optional role-ref element is set, it is expected that the support activity will act for every single user in the specified role(s). That is: the same support activity is repeated for every user in the role(s). When the role-ref is not available, the support activity is a single activity (like the learning-activity).

Multiplicity: The <support-activity> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **isvisible (mandatory).** True or false, initial visibility attribute (see IMS CP). Data type = token group.
- **parameters (optional).** See IMS CP. Data type = string.

Elements:

- title
- role-ref
- environment-ref
- activity-description
- complete-activity
- on-completion
- metadata

Example:

```
<support-activity identifier="Support_participants_preparation">
  <title>Support Participants Perparation</title>
  <activity-description>
    <item identifier="Support_participants_preparation_AD_res"/>
  </activity-description>
</support-activity>
```

4.4.3 <activity-structure>

Description: An activity-structure groups activities in sequences or selections. The tree is handled depth first (and not breadth first). If there is a sequencing element it overrules the Attributes: structure-type and number-to-select.

Multiplicity: The <activity-structure> element is mandatory and occurs only 1 time.

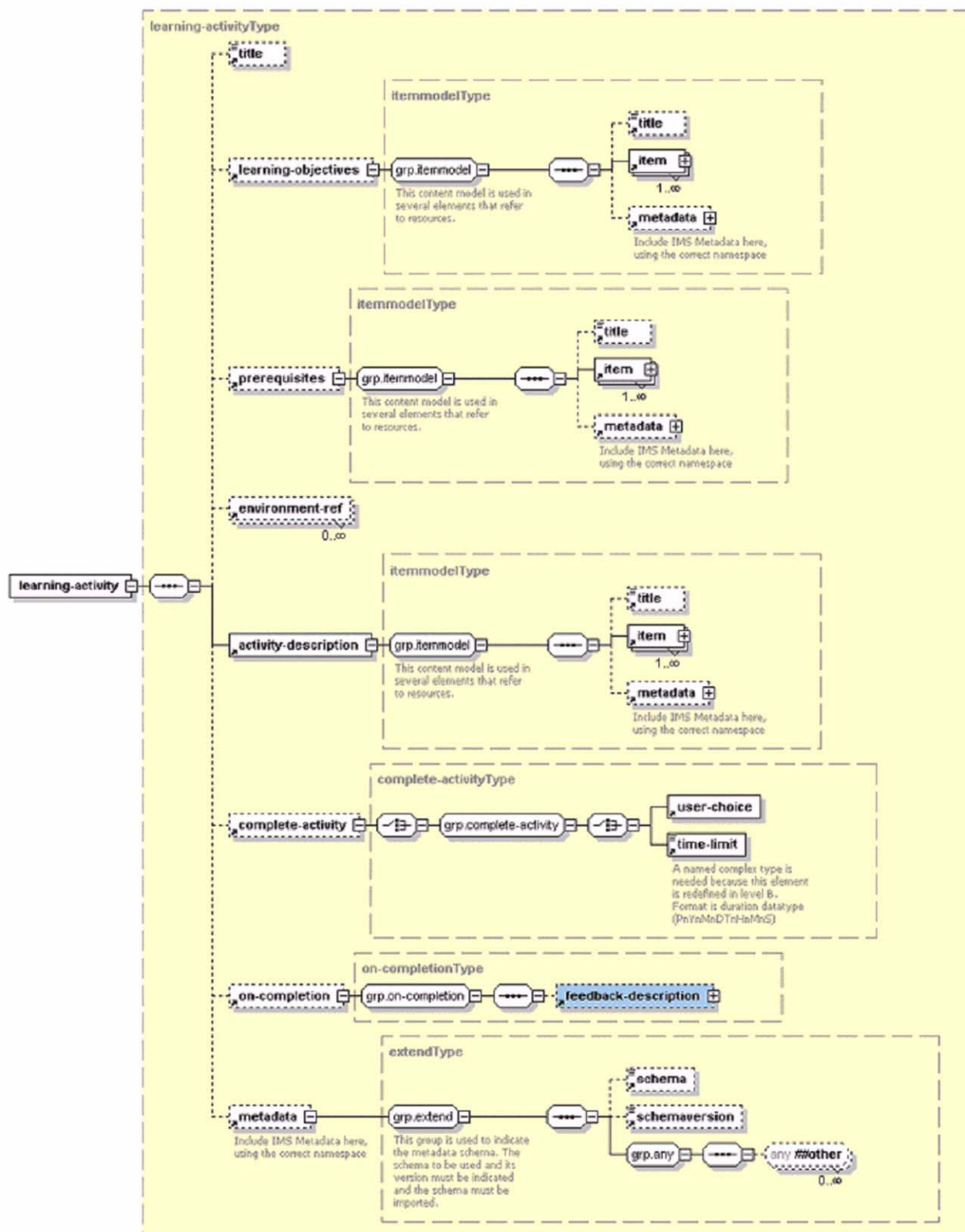
Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **number-to-select (optional).** When the attribute 'number-to-select' is set, the activity-structure is completed when the total number of activities selected are completed. The number-to-select must be similar or smaller than the number of activities (including unit-of-learnings) which are at the immediate child level. When the number-to-select is not set, the activity-structure is completed when all the activities in the structure are completed. Data type = string.
- **sort (optional).** The attribute 'sort' determines the sort-order in relation to the visibility. Default the order in which activities are made visible is in the order specified in the activity-structure. Data type = token group.
- **structure-type (optional).** Indicates whether the activity-structure represents a sequence or a selection. Data type = token group.

Example:

```
<activity-structure identifier="Preparation" structure-type="sequence">
  <title>Preparation the Versailles Negotiations</title>
  <learning-activity-ref ref="Preparation_Intro"/>
  <!-- The next item references another activity-structure. See next -->
  <learning-activity-ref ref="Objectives_Background"/>
  <learning-activity-ref ref="Offline_Strategy"/>
  <learning-activity-ref ref="Offline_Negotiation"/>
</activity-structure>
```

4.5 <learning-activity> Elements



4.5.1 <title>

See <title> above.

4.5.2 <learning-objectives>

Description: Learning objectives describe the intended outcome for learners. Learning-objectives & prerequisites contain a standard organization of items, referring to resources or sub manifests. Resource type connected to learning objectives and prerequisites can be webcontent, imslcontent or it can point to an IMS Reusable Competency Definition Schema. There are two locations where learning-objectives & prerequisites are specified: - At the level of the learning design (in the root of learning-design) - At the level of learning-activities (within learning-activities). The first ones are a more general description; the second ones are more concrete. There are two types of learning-objectives: -1- human readable descriptions (the items point to text resources) -2- machine-readable specifications. These are addressed through the href attribute of the resources pointed at. The learning-objectives schema's could be user-defined or fixed by an organization. In the latter case, the texts of the learning objectives are referred to (through href).

Multiplicity: The <learning-objectives> element is optional and occurs zero or 1 time.

Attributes:

- **{itemmodel} (mandatory).** See IMS CP. A Schema group. Data type = group.

Elements:

- title
- item
- metadata

Example:

```
<learning-objectives>
  <item isvisible="true" parameters="" identifierref="" identifier="LOB_learning_objectives"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"></item>
</learning-objectives>
```

4.5.3 <prerequisites>

See <prerequisites> above.

4.5.4 <environment-ref>

Description: Refers to an environment in this package.

Multiplicity: The <environment-ref> element is optional and occurs zero or more times.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

Elements:

Example:

```
<activity-structure identifier="AS_lessons_and_procedure" number-to-select="2"
structure-type="sequence">
  <title e-dtype="string" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
    <environment-ref ref="E_interactive_electronic_training_manual"/>
    <activity-structure-ref ref="AS_fuel_valve_lessons"/>
    <activity-structure-ref ref="AS_fuel_valve_removal_procedure"/>
</activity-structure>
```


4.5.5 <activity-description>

Description: Alias: task. The activity-description is the actual cue given to the user (rendered in the user-interface) to describe the activity to be performed by the user. In most cases the activity-description is a text (of type webcontent or imslcontent). In other cases it can be an audio-file (webcontent), a video file or any other cue to the user.

Activity-descriptions also define the environment for the activity. Every noun mentioned in the description refers to a resource in the environment. It is up to the author to have a strict representation of the nouns in the environment or a more open one (leaving nouns implicit).

Multiplicity: The <activity-description> element is mandatory and occurs only 1 time.

Attributes:

- **{itemmodel} (mandatory).** See IMS CP. A Schema group. Data type = group.

Elements:

- title
- item
- metadata

Example:

```
<activity-description>
  <item isvisible="true" parameters="" identifierref="" identifier="I_fuel_valve_lesson_intro"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"/>
</activity-description>
```

4.5.6 <complete-activity>

Description: Contains a choice of elements to specify when an activity is completed. When this element does not occur, the activity is set to 'completed'.

Multiplicity: The <complete-activity> element is optional and occurs zero or 1 time.

Elements:

- user-choice
- time-limit

Example:

```
<complete-activity>
  <time-limit/>
</complete-activity>
```

4.5.6.1 <user-choice>

Description: This element is used in the completed element of activities and specifies that the user may decide him or her self when the activity is completed. This means that a control must be available in the user-interface to set the activity status to 'completed'. A user can do this once (no undo). When he/she selected the activity to be completed, than this activity stays completed in the run.

Multiplicity: The <user-choice> element is mandatory and occurs only 1 time.

4.5.6.2 <time-limit>

Description: The time limit specifies that it is completed when a certain amount of time have been passed, relative to the start of the run of the current unit of learning. The data type time is expressed in the 'duration' format (explained elsewhere in the information model). The time is always counted relative to the time when the run of the unit-of-learning has been started (see element: 'time-unit-of-learning-started'. Authors have to take care that the time

limits set on role-parts, acts and plays are logical. In runtime the time limit of the play overrules the time limit on act and that one on the role-parts. The time-limit may be specified in a property (property-ref attribute, of type loc-property, datatype=string, to be declared by the author). In that case an author may set controls (set-property) on this property for users to control the value of the property. When a property-ref is specified, content in the element is ignored: the property overrules.

Multiplicity: The <time-limit> element is mandatory and occurs only 1 time.

Attributes:

- **property-ref (optional).** Refers to a property identifier. Data type = idref.

4.5.7 <on-completion>

Description: When an activity, act, play or unit-of-learning is completed, the optional actions contained in this element are executed. In level A it contains only one element. The wrapper is available for the extensions of level B and C.

Multiplicity: The <on-completion> element is optional and occurs zero or 1 time.

Elements:

- feedback-description

4.5.7.1 <feedback-description>

Description: The underlying item elements points to a resource (of type webcontent or imslcontent), where the feedback description can be found. After completion this text becomes visible.

Multiplicity: The <feedback-description> element is optional and occurs zero or 1 time.

Attributes:

- **{itemmodel} (mandatory).** See IMS CP. A Schema group. Data type = group.

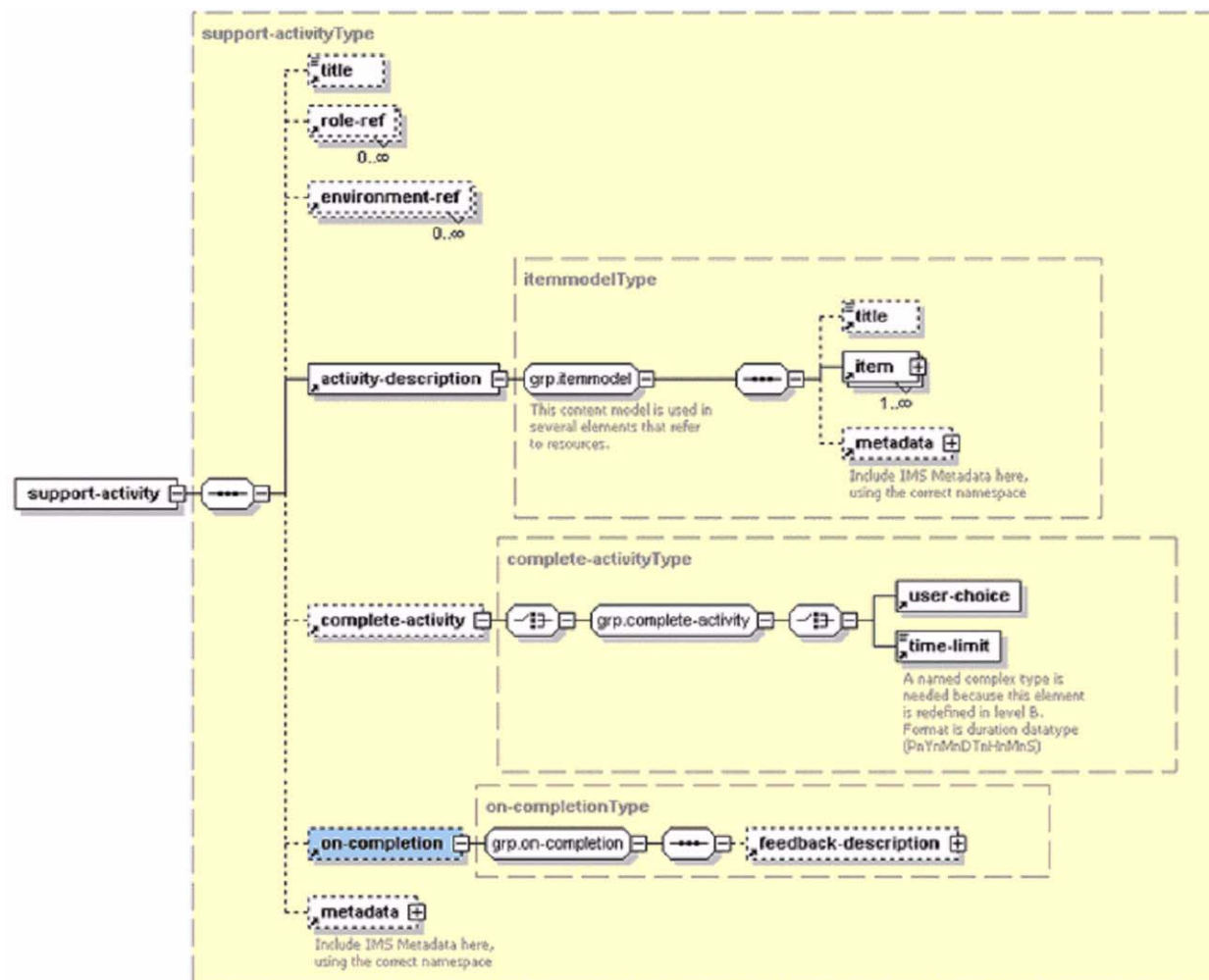
Elements:

- title
- item
- metadata

4.5.8 <metadata>

See <metadata> above.

4.6 <support-activity> Elements



4.6.1 <title>

See <title> above.

4.6.2 <role-ref>

Description: Role refers to the identifier of the resource of the role. The element can be used as an operand in an expression.

Multiplicity: The <role-ref> element is optional and occurs zero or more times.

Attributes:

- ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.6.3 <environment-ref>

See <environment-ref> above.

4.6.4 <activity-description>

See <activity-description> above.

4.6.5 <complete-activity>

See <complete-activity> above.

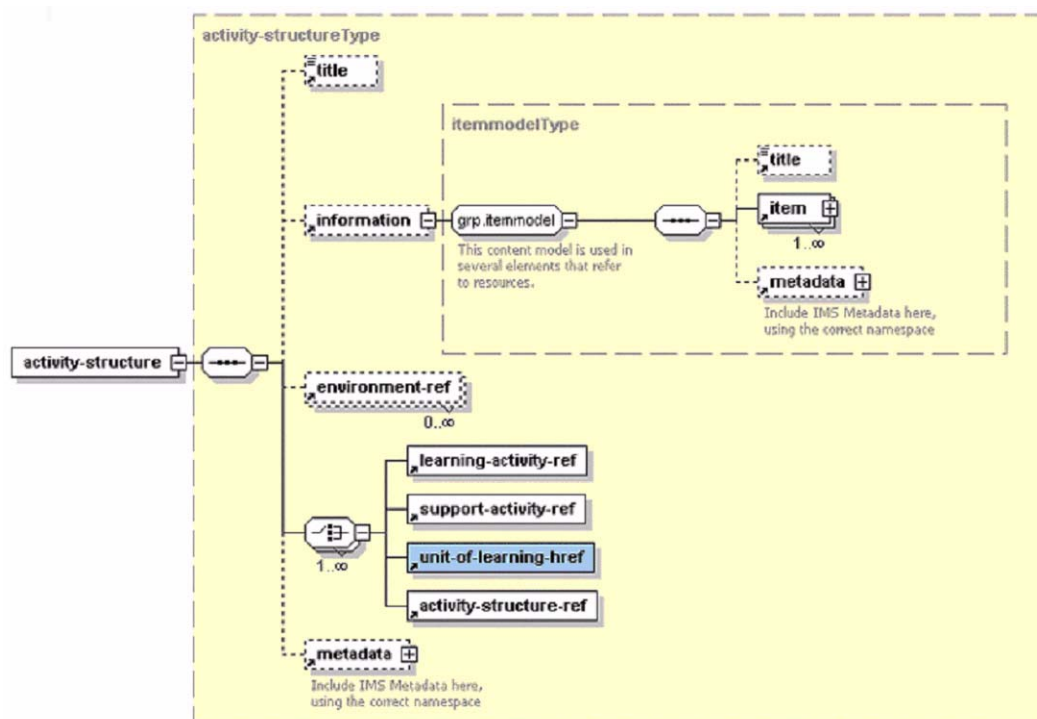
4.6.6 <on-completion>

See <on-completion> above.

4.6.7 <metadata>

See <metadata> above.

4.7 <activity-structure> Elements



4.7.1 <title>

See <title> above.

4.7.2 <information>

Description: The <information> element specifies the set of items pointing to the resource(s) where the information can be found.

Multiplicity: The <information> element is optional and occurs zero or 1 time.

Attributes:

- **{itemmodel} (mandatory).** See IMS CP. A Schema group. Data type = group.

Elements:

- title
- item
- metadata

4.7.3 <environment-ref>

See <environment-ref> above.

4.7.4 <learning-activity-ref>

Description: Refers to a learning-activity. The element can be used as an operand in a calculation or expression.

Multiplicity: The <learning-activity-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.7.5 <support-activity-ref>

Description: Refers to a support-activity. The element can be used as an operand in a calculation or expression.

Multiplicity: The <support-activity-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.7.6 <unit-of-learning-href>

Description: The element can be used as an operand in a calculation or expression. This element refers to the resource of a unit-of-learning (uol). That can be the resource of this current unit-of-learning, the resource of a unit-of-learning present in the package or a resource pointing to a unit-of-learning outside of the package (absolute URI expressed as an ID).

Multiplicity: The <unit-of-learning-href> element is mandatory and occurs only 1 time.

Attributes:

- **href (mandatory).** Refers to a URI. Data type = string.

4.7.7 <activity-structure-ref>

Description: Reference to an activity-structure.

Multiplicity: The <activity-structure-ref> element is mandatory and occurs only 1 time.

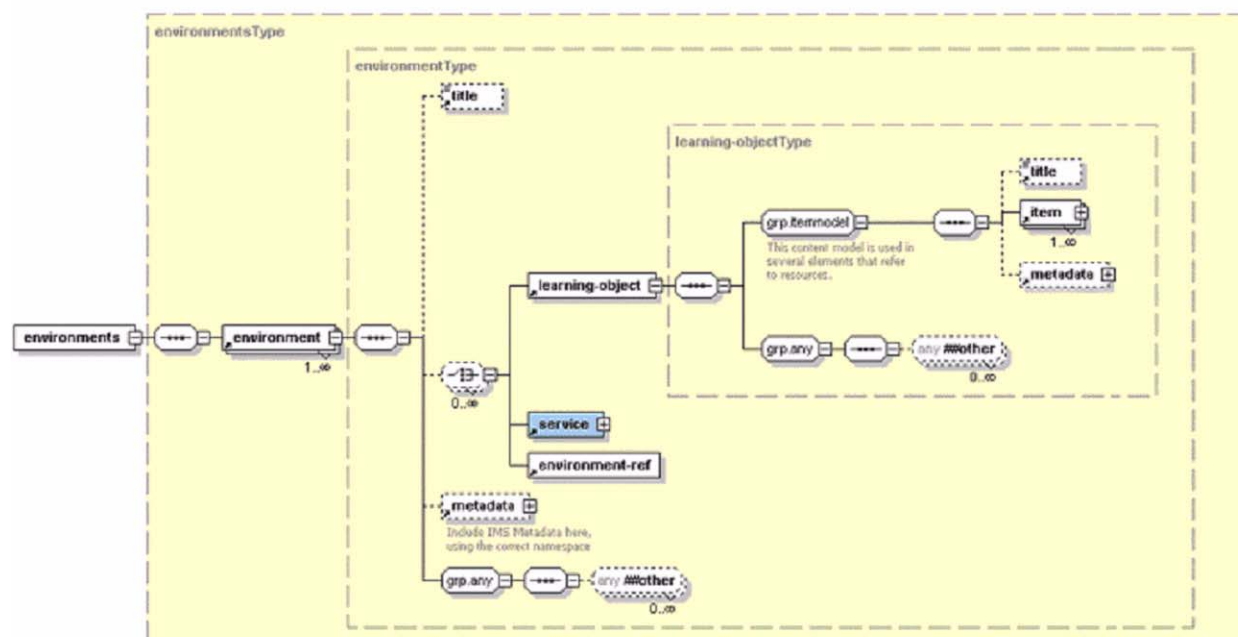
Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.7.8 <metadata>

See <metadata> above.

4.8 <environments> Elements



4.8.1 <environment>

Description: Contains a sequence of elements to model an environment.

Multiplicity: The <environment> element is mandatory and occurs 1 or more times.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- learning-object
- service
- environment-ref
- metadata

Example:

```
<environment identifier="Versailles_Aims">
  <title>Aims of the Versailles Experience</title>
  <learning-object identifier="Versailles_Aims_LO">
    <item identifier="V_Aims_item" identifierref="V_Aims_res"/>
  </learning-object>
</environment>
```

4.8.1.1 <title>

See <title> above.

4.8.1.2 <learning-object>

Description: Learning objects are reflected with an included schema (e.g., IMS QTI) or can reference resources through the item elements.

Multiplicity: The <learning-object> element is mandatory and occurs only 1 time.

Attributes:

- **class (optional).** The class attribute refers to the value of class attributes available in learning-design or content elements. Contains a CDATA string. Just as in HTML more than one class may be specified in one CDATA string, each separated with a blank space. The priority order for classes is the same as specified in the CSS specification (see <http://www.w3.org/style/css>). Any element can in principle have the class attribute. 'Class' is a global defined W3C attribute for html 4.0 and XHTML [LD14]. This attribute assigns a class name or set of class names to an element. Any number of elements may be assigned the same class name or names. Multiple class names must be separated by white space characters. The class element can be used for semantic grouping of elements and be manipulated by IMSLD conditions and stylesheets. When sending a learning object to a web client, include the class attribute and value(s). Data type = string.
- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **isvisible (mandatory).** True or false, initial visibility attribute (see IMS CP). Data type = token group.
- **parameters (optional).** See IMS CP. Data type = string.
- **type (optional).** The type of learning object (e.g. knowledge-object, tool-object test-object). Vocabulary used can be the one of 'learning resource type' element from the IEEE LTSC LOM. Data type = string.

Elements:

- title
- item
- metadata
- ###other

4.8.1.2.1 <title>

See <title> above.

4.8.1.2.2 <item>

See <item> above.

4.8.1.2.3 <metadata>

See <metadata> above.

4.8.1.2.4 ###other

Description: The ###other construct is provided as an extension mechanism to allow additional elements to be added below <learning-object>. These additional elements would be defined by a modified schema or namespaced extension.

Multiplicity: Any number of additional elements may be added.

4.8.1.3 <service>

Description: A service is a declaration of a service facility which has to be bound during instantiation of a run of a unit of learning. To automate the set up process of a service facility from a service declaration, the run time data from the instantiated learning design would be translated into a configuration format used by the conference system if the conference is to be automatically set up. This is an implementation issue. It is also possible that a system manager could read this information and set up the required conference space manually, but the intent is to alleviate the manager of this task by enabling it to be automated. The service specification is extensible by namespacing in additional services.

When instantiating a service, the runtime systems needs to maintain a handle on the ‘context’ to which the service is to be bound and determine the users to whom the service is being made available. A service will be referenced by an item element’s identifierref attribute. The item will be within an environment. The environment in turn will be associated with an activity, or possibly directly with a role-part, associating it with a role. The activity or role-part forms the context for the use of the service. The users playing the role are those that have access to the service.

Multiplicity: The <service> element is mandatory and occurs only 1 time.

Attributes:

- **class (optional).** The class attribute refers to the value of class attributes available in learning-design or content elements. Contains a CDATA string. Just as in HTML more than one class may be specified in one CDATA string, each separated with a blank space. The priority order for classes is the same as specified in the CSS specification (see <http://www.w3.org/style/css>). Any element can in principle have the class attribute. ‘Class’ is a global defined W3C attribute for html 4.0 and XHTML [LD14]. This attribute assigns a class name or set of class names to an element. Any number of elements may be assigned the same class name or names. Multiple class names must be separated by white space characters. The class element can be used for semantic grouping of elements and be manipulated by IMSLD conditions and stylesheets. When sending a learning object to a web client, include the class attribute and value(s). Data type = string.
- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **invisible (mandatory).** True or false, initial visibility attribute (see IMS CP). Data type = token group.
- **parameters (optional).** See IMS CP. Data type = string.

Elements:

- send-mail
- conference
- index-search

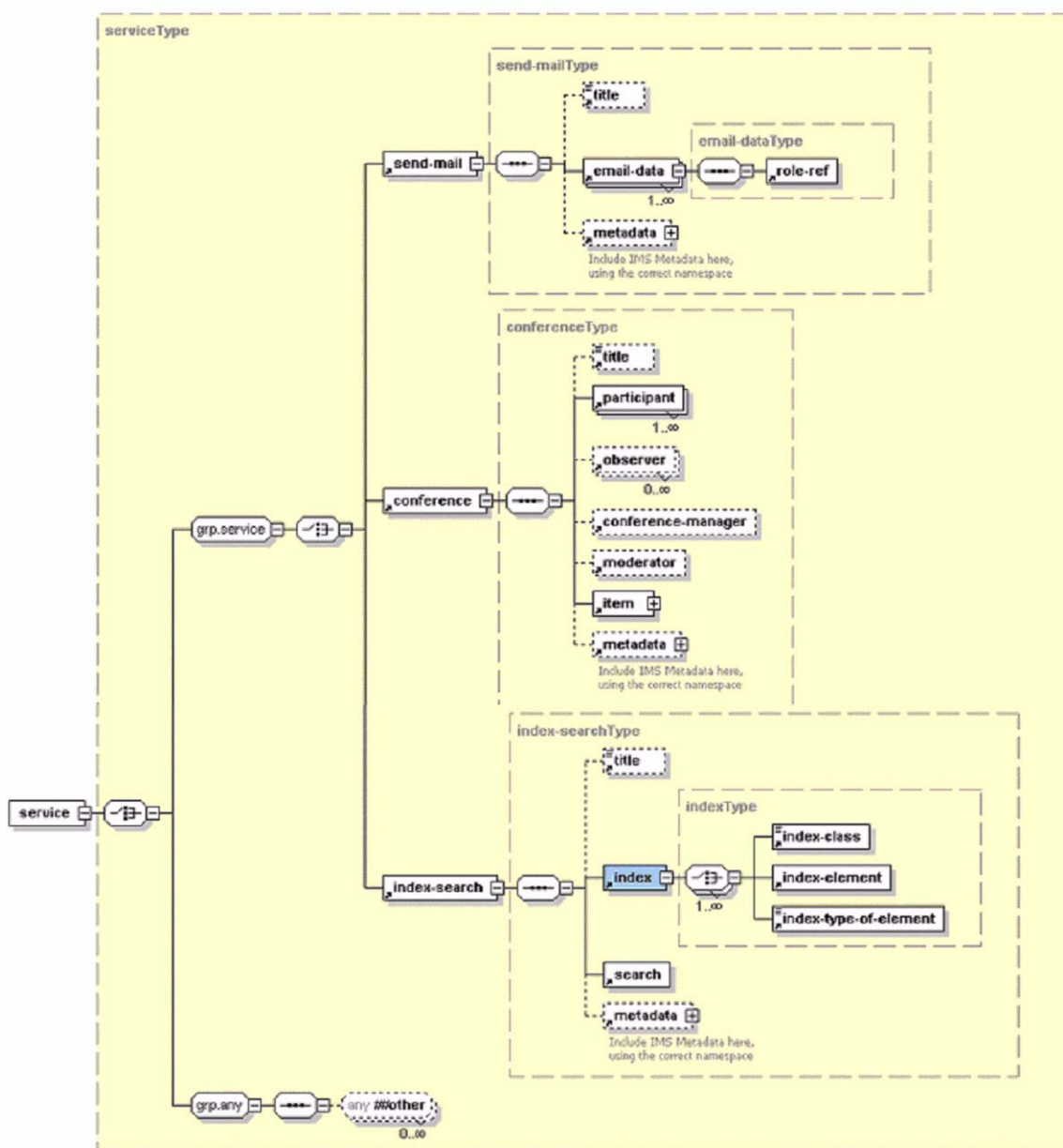
4.8.1.4 <environment-ref>

See <environment-ref> above.

4.8.1.5 <metadata>

See <metadata> above.

4.9 <service> Elements



4.9.1 <send-mail>

Description: This service is used to send mail to users in roles (with mail address in property for level b/c).

Multiplicity: The `<send-mail>` element is mandatory and occurs only 1 time.

Attributes:

- **select (mandatory).** Fixed choice: 'all-persons-in-role' or 'persons-in-role'. With the first choice the user agent only allows to send messages to the role, indicating that all persons in the role get the message. With the second choice a user gets the possibility to select one or more individuals within the specified role to send the message to. Data type = token group.

Elements:

- title
- email-data
- metadata

Example:

```
<imsld:send-mail select="all-persons-in-role">
  <imsld:email-data>
    <imsld:role-ref ref="Recorder" />
  </imsld:email-data>
</imsld:send-mail>
```

4.9.1.1 <title>

See <title> above.

4.9.1.2 <email-data>

Description: This is used for send-mail purposes (as a service in the environment, or in notifications). In level B, the properties on this element refer to the property resources where the relevant e-mail data can be found for the connected role. In level A, the source is not specified explicitly and is left to implementers to decide how to address the data needed. Both properties (email, username) should be available for all persons assigned to the role and the sending party.

Multiplicity: The <email-data> element is mandatory and occurs 1 or more times.

Elements:

- role-ref

4.9.1.2.1 <role-ref>

Description: Role refers to the identifier of the resource of the role. The element can be used as an operand in an expression.

Multiplicity: The <role-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.9.1.3 <metadata>

See <metadata> above.

4.9.2 <conference>

Description: The elements participant, observer, conference-manager, moderator facilitate the setting of the user rights in the conferences. It depends on the implementation how this is managed: 1. When the conference system is an integral part of the runtime it is expected to be set automatically; 2. When the conference is external the user-rights can be set manually by the conference manager. The conference managers, must be able to get a list from the runtime agent about which conferences of what type, for what users with what rights must be set. 3. The latter can also be implemented by a developing a legacy interface to the rights management system of the conferencing system. In all instances the runtime system must be able to provide this information in a structured way. The item element refers to the resource where the conferencing system is to be found or identified. External conferencing systems can be of any kind accessible through the internet (resource type is webcontent). Examples: netmeeting, placeware (synchronous), first-class, lotus notes, news groups (asynchronous). An announcement object sets the rights: creator of announcement = participant. Reader of announcements = observer.

Multiplicity: The <conference> element is mandatory and occurs only 1 time.

Attributes:

- **conference-type (mandatory).** Fixed choice to specify the type of conference facility that is expected to be present at runtime: synchronous, asynchronous or announcement. Data type = token group.

Elements:

- title
- participant
- observer
- moderator
- conference-manager
- item
- metadata

Example:

```
<imsld:conference conference-type="asynchronous">
  <imsld:participant role-ref="Learner"/>
  <imsld:participant role-ref="Teacher"/>
  <imsld:participant role-ref="Expert"/>
  <!-- NB MODERATOR -->
  <imsld:moderator role-ref="Chair"/>
  <item identifier="" />
</imsld:conference>
```

4.9.2.1 <title>

See <title> above.

4.9.2.2 <participant>

Description: Specifies who the participants are in the conference. Participants can read (listen/see) the information, and can contribute to the conference. This element has an effect on setting the user rights in the conference. At least one role must be specified to identify the participants in the conference.

Multiplicity:**Attributes:**

- **role-ref (mandatory).** Refers to a role identifier. Data type = idref.

4.9.2.3 <observer>

Description: Specifies who the observers are in the conference. Observers have only reading rights; they may not contribute. This element has an effect on setting the user rights in the conference.

Multiplicity: The <observer> element is optional and occurs zero or more times.

Attributes:

- **role-ref (mandatory).** Refers to a role identifier. Data type = idref.

4.9.2.4 <moderator>

Description: Specifies who the moderators are in the conference. Moderators are persons who have the right to control and change the contributions of participants before they are made visible to other participants or observers. When a moderator is specified it means that participants may not contribute directly to the conference, but via the moderator. The moderator can reject, adapt or accept a proposed contribution of a participant. In all cases the contributor is

notified of the judgement of the moderator. When there are more users in the role connected to the moderator, all have the same rights, but always the first one who did the job decides. This element has an effect on the setting of the user rights in the conference.

Multiplicity: The <moderator> element is optional and occurs zero or more times.

Attributes:

- **role-ref (mandatory).** Refers to a role identifier. Data type = idref.

4.9.2.5 <conference-manager>

Description: The conference manager is allowed to create new sub conferences and delete conferences he/she created. The new conferences are children of the existing base conference. The conference manager may not delete the base conference. It is deleted by system management when deleting the information of the (completed) run of the unit of learning. The conference manager has all the rights of observer, participant.

Multiplicity: The <conference-manager> element is optional and occurs zero or 1 time.

Attributes:

- **role-ref (mandatory).** Refers to a role identifier. Data type = idref.

4.9.2.6 <item>

Description: See IMS CP.

Multiplicity: The <item> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **identifierref (mandatory).** Refers to an identifier of a source in the conten package (outside the learning design). Data type = idref.
- **isvisible (optional).** True or false, initial visibility attribute (see IMS CP). Data type = string.
- **parameters (optional).** See IMS CP. Data type = string.

Elements:

- title
- item
- metadata

4.9.2.6.1 <title>

See <title> above.

4.9.2.6.2 <item>

See <item> above.

4.9.2.6.3 <metadata>

See <metadata> above.

4.9.2.7 <metadata>

See <metadata> above.

4.9.3 <index-search>

Description: Contains a sequence of elements that declare an index and/or search service facility.

Multiplicity: The <index-search> element is mandatory and occurs only 1 time.

Elements:

- title
- index
- search
- metadata

4.9.3.1 <title>

See <title> above.

4.9.3.2 <index>

Description: A choice of elements to specify indexing aspects, used to set up a search service. The index is made in the background (not visible to users). The visibility is determined with the search element. The functionality of the index is dependent on the search element: - When search is free-text-search, then the index is made on the resource pointed at in the index (i.e. the underlying html texts). - When search is index-with/without-reference, than only an index is made of the elements which share the same class, including underlying items. This has the form of a table of content.

Multiplicity: The <index> element is mandatory and occurs only 1 time.

Elements:

- index-class
- index-element
- index-type-of-element

4.9.3.2.1 <index-class>

Description: This element selects the class to make the index on. Only one class item per element may be provided. Example: <index-class index-class="problemdescription"/> makes an index on all objects in the design which have one of the strings in the class attribute assigned to "problemdescription".

Multiplicity: The <index-class> element is mandatory and occurs only 1 time.

4.9.3.2.2 <index-element>

Description: This element selects the element to make the index on. The index attribute specifies the element to index-on (only one reference per index-element). This indexing only makes sense when there is a structure to index on, or underlying text to index for free-text-search.

Multiplicity: The <index-element> element is mandatory and occurs only 1 time.

Attributes:

- **index (mandatory).** Refers to the element to make the index of. Data type = idref.

4.9.3.2.3 <index-type-of-element>

Description: In this element the type of element to index on is entered. Only one element name per index-type-element occurrence. The element names much match the element names uses in the IMSLD schema. For example: <index-type-of-element>learning-activity</index-type-of-element>

Multiplicity: The <index-type-of-element> is mandatory and occurs only 1 time.

4.9.3.3 <search>

Description: This element specifies how a user can access the indexed entities. There are three possibilities: 1. The user gets a free text search dialog, where he can search the index in a free text format (this also means that the index has to be build for free text retrieval). The syntax for free text retrieval is implementation dependent, e.g. the format found in search engines like Google or Yahoo. 2. The user is presented a text index (table of content) with (hyper-) linked (or on other media e.g. page numbers) references to the source. 3. The user is presented a text index (table of content) without (hyper-) linked references. This provides e.g. information about the structure of the unit of learning.

Multiplicity: The <search> element is mandatory and occurs only 1 time.

Attributes:

- **search-type (mandatory).** Fixed choice to indicate the type of search facility that is expected at runtime: free-text-search, index-with-reference, index-without-reference. A free text search uses a free text search index. An index without reference is a list of terms without page-numbers or hyperlinks. An index with references use page-numbers or hyperlinks (depending on the publication medium used). Data type = token group.

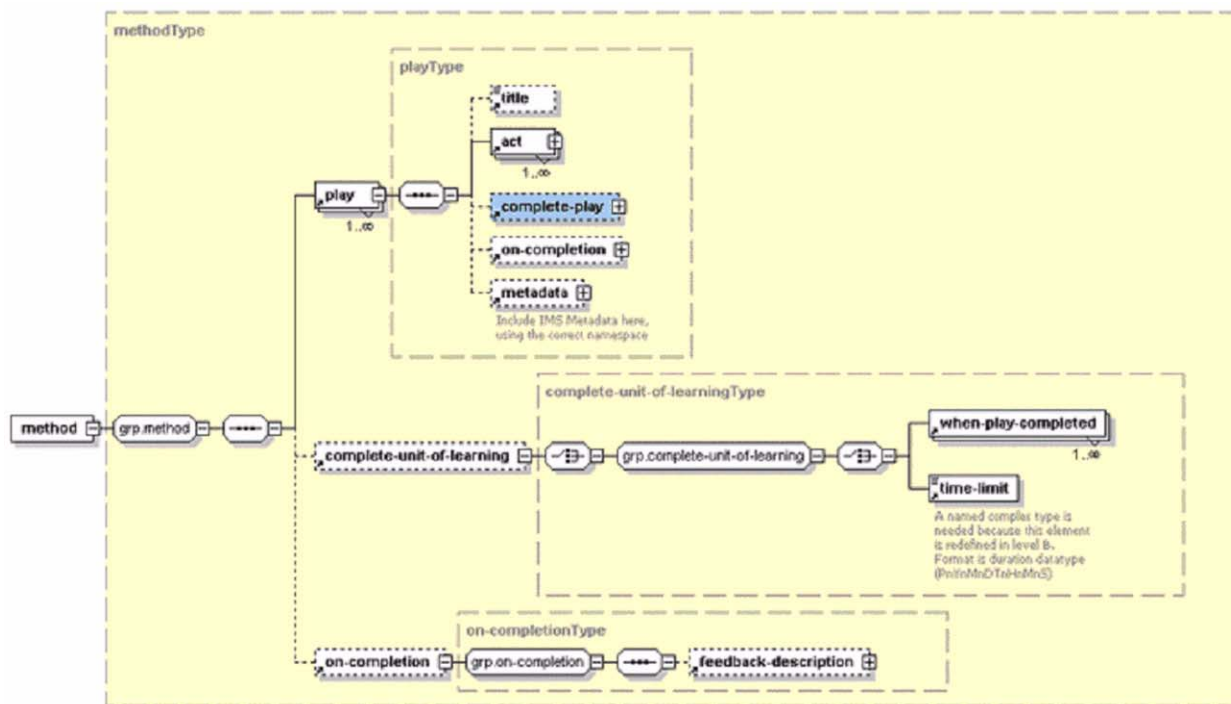
4.9.3.4 <metadata>

See <metadata> above.

4.9.4 ###other

See ###other above.

4.10 <method> Elements



4.10.1 <play>

Description: The play is the root element when interpreting the learning design. It represents the flow of activities during the learning process (the ‘workflow’ or better: the ‘learningflow’). A play consists of a series of acts and an act consists of a series of role-part’s. There is always at least one play in every learning design (and every unit-of-learning). In runtime the play is interpreted to show and hide activities, (other)units-of-learning, environments and resources to the users. When there is more than one play, these are interpreted concurrently and independent of each other. The same user can see the results of more than one play in the user-interface. Practical experience has shown that a lot of designs use multiple plays, to represent the flows of activities per role, e.g. a play for the learners and a play for staff. However this can only be done when the activities are independent of each other.

Multiplicity: The <play> element is mandatory and occurs 1 or more times.

Attributes:

- **identifier (optional).** An identifier that is unique within the learning design file (ID). Data type = ID.
- **isvisible (mandatory).** True or false, initial visibility attribute (see IMS CP). Data type = token group.

Elements:

- title
- act
- complete-play
- on-completion
- metadata

Example:

```
<play identifier="P_Boeing_simplified" isvisible="true">
  <act identifier="ACT_individualized_learning">
    <role-part identifier="RP_individualized_learning">
      <role-ref ref="R_learner"/>
      <activity-structure-ref ref="AS_boeing_simplified" />
    </role-part>
    <complete-act>
      <when-role-part-completed ref="RP_individualized_learning"/>
    </complete-act>
  </act>
  <complete-play>
    <when-last-act-completed/>
  </complete-play>
</play>
```

4.10.2 <complete-unit-of-learning>

Description: A choice of elements to specify when a unit-of-learning is completed. When this element does not occur, the completed is set to ‘unlimited’.

Multiplicity: The <complete-unit-of-learning> element is optional and occurs zero or 1 time.

Elements:

- when-play-completed
- time-limit

4.10.2.1 <when-play-completed>

Description: This element states that an unit-of-learning is completed when the referenced play(s) is (are) completed. More than one play can be selected, meaning that all the referenced plays must be completed before the unit-of-learning is completed. When a unit-of-learning is completed this should be made aware in the runtime environment to the managers of the system.

Multiplicity: The <when-play-completed> element is mandatory and occurs 1 or more times.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

4.10.2.2 <time-limit>

Description: The time limit specifies that it is completed when a certain amount of time have been passed, relative to the start of the run of the current unit of learning. The data type time is expressed in the 'duration' format (explained elsewhere in the information model). The time is always counted relative to the time when the run of the unit-of-learning has been started (see element: 'time-unit-of-learning-started'. Authors have to take care that the time limits set on role-parts, acts and plays are logical. In runtime the time limit of the play overrules the time limit on act and that one on the role-parts. The time-limit may be specified in a property (property-ref attribute, of type loc-property, datatype=string, to be declared by the author). In that case an author may set controls (set-property) on this property for users to control the value of the property. When a property-ref is specified, content in the element is ignored: the property overrules.

Multiplicity: The <time-limit> element is mandatory and occurs only 1 time.

4.10.3 <on-completion>

Description: When an activity, act, play or unit-of-learning is completed, the optional actions contained in this element are executed. In level A it contains only one element. The wrapper is available for the extensions of level B and C.

Multiplicity: The <on-completion> element is optional and occurs zero or 1 time.

Elements:

- feedback-description

4.10.3.1 <feedback-description>

Description: The underlying item elements points to a resource (of type webcontent or imslidcontent), where the feedback description can be found. After completion this text becomes visible.

Multiplicity: The <feedback-description> element is optional and occurs zero or 1 time.

Elements:

- title
- item
- metadata

4.10.3.1.1 <title>

See <title> above.

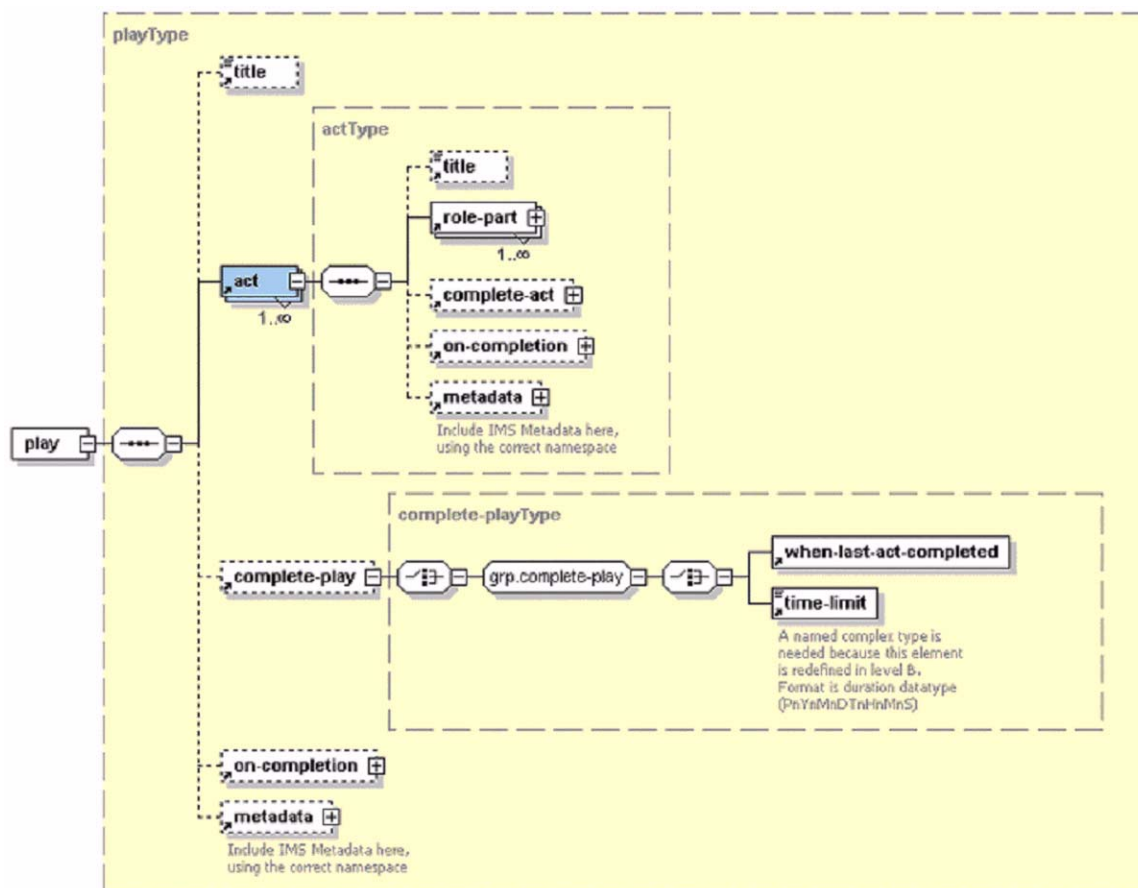
4.10.3.1.2 <item>

See <item> above.

4.10.3.1.3 <metadata>

See <metadata> above.

4.11 <play> Elements



4.11.1 <title>

See <title> above.

4.11.2 <act>

Description: A play consists of a series of acts and an act consists of a series of role-parts. An act represents a series of concurrent role-parts. There is at least one act in a play. When there is more than one act in a play these are interpreted in a sequenced manner: from first act to last act. Only one act in a play is the active act at any moment in time, starting with the first. When the first act is completed, the second act is made the active act. The first is still visible and accessible, but in the interface it is made clear that it is only looking in history. When the second act is completed, the third act is made active, etc. Acts, which are sequenced in row after the current active act, are never visible. Conditions cannot overrule this, meaning that the act is of higher priority than conditions.

Multiplicity: The `<act>` element is mandatory and occurs 1 or more times.

Attributes:

- identifier (optional).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title

- role-part
- complete-act
- on-completion
- metadata

Example:

```
<act>
  <role-part>
    <role-ref ref="Learner"/>
    <learning-activity-ref ref="Versailles_Overview"/>
  </role-part>
  <role-part>
    <role-ref ref="Support_Staff"/>
    <learning-activity-ref ref="Versailles_Overview"/>
  </role-part>
</act>
```

4.11.3 <complete-play>

Description: A choice of elements to specify when a play is completed. When this element does not occur, the completed is set to 'unlimited'.

Multiplicity: The <complete-play> element is optional and occurs zero or more times.

Elements:

- when-last-act-completed
- time-limit

4.11.3.1 <when-last-act-completed>

Description: This element states that a play is completed when the last act is completed.

Multiplicity: The <when-last-act-completed> element is mandatory and occurs only 1 time.

4.11.3.2 <time-limit>

See <time-limit> above.

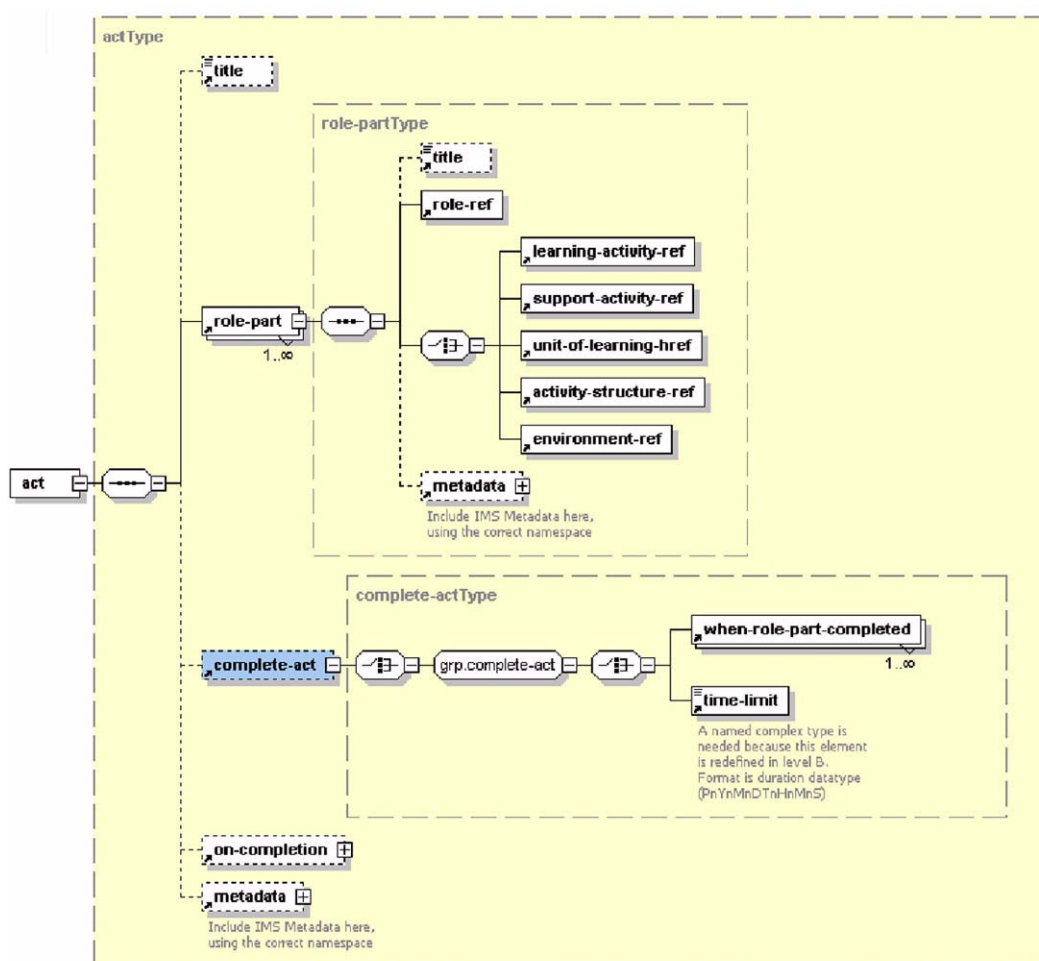
4.11.4 <on-completion>

See <on-completion> above.

4.11.5 <metadata>

See <metadata> above.

4.12 <act> Elements



4.12.1 <title>

See <title> above.

4.12.2 <role-part>

Description: A play consists of a series of acts and an act consists of a series of role-parts. A role-part relates exactly one role to exactly one type of activity (including the performance of another unit-of-learning and activity-structures). Role-parts within one act, are performed concurrently. When an activity or item attribute is visible is set to 'false', the link in the activity-tree may be made visible when the role-part sets the activity for a role (implementation dependent), but the content is not accessible.

Multiplicity: The `<role-part>` element is mandatory and occurs 1 or more times.

Attributes:

- identifier (optional).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- role-ref
- learning-activity-ref

- support-activity-ref
- unit-of-learning-href
- activity-structure-ref
- environment-ref
- metadata

Example:

```
<role-part>
  <role-ref ref="Teacher"/>
  <learning-activity-ref ref="Preparation"/>
</role-part>
```

4.12.2.1 <title>

See <title> above.

4.12.2.2 <role-ref>

See <role-ref> above.

4.12.2.3 <learning-activity-ref>

See <learning-activity-ref> above.

4.12.2.4 <support-activity-ref>

See <support-activity-ref> above.

4.12.2.5 <unit-of-learning-ref>

See <unit-of-learning-ref> above.

4.12.2.6 <activity-structure-ref>

See <activity-structure-ref> above.

4.12.2.7 <environment-ref>

See <environment-ref> above.

4.12.2.8 <metadata>

See <metadata> above.

4.12.3 <complete-act>

Description: A choice of elements to specify when an act is completed. When this element does not occur, the completed is set to 'unlimited'.

Multiplicity: The <complete-act> element is optional and occurs zero or 1 time.

Elements:

- when-role-part-completed
- time-limit

4.12.3.1 <when-role-part-completed>

See <when-role-part-completed> above.

4.12.3.2 <time-limit>

See <time-limit> above.

4.12.4 <on-completion>

See <on-completion> above.

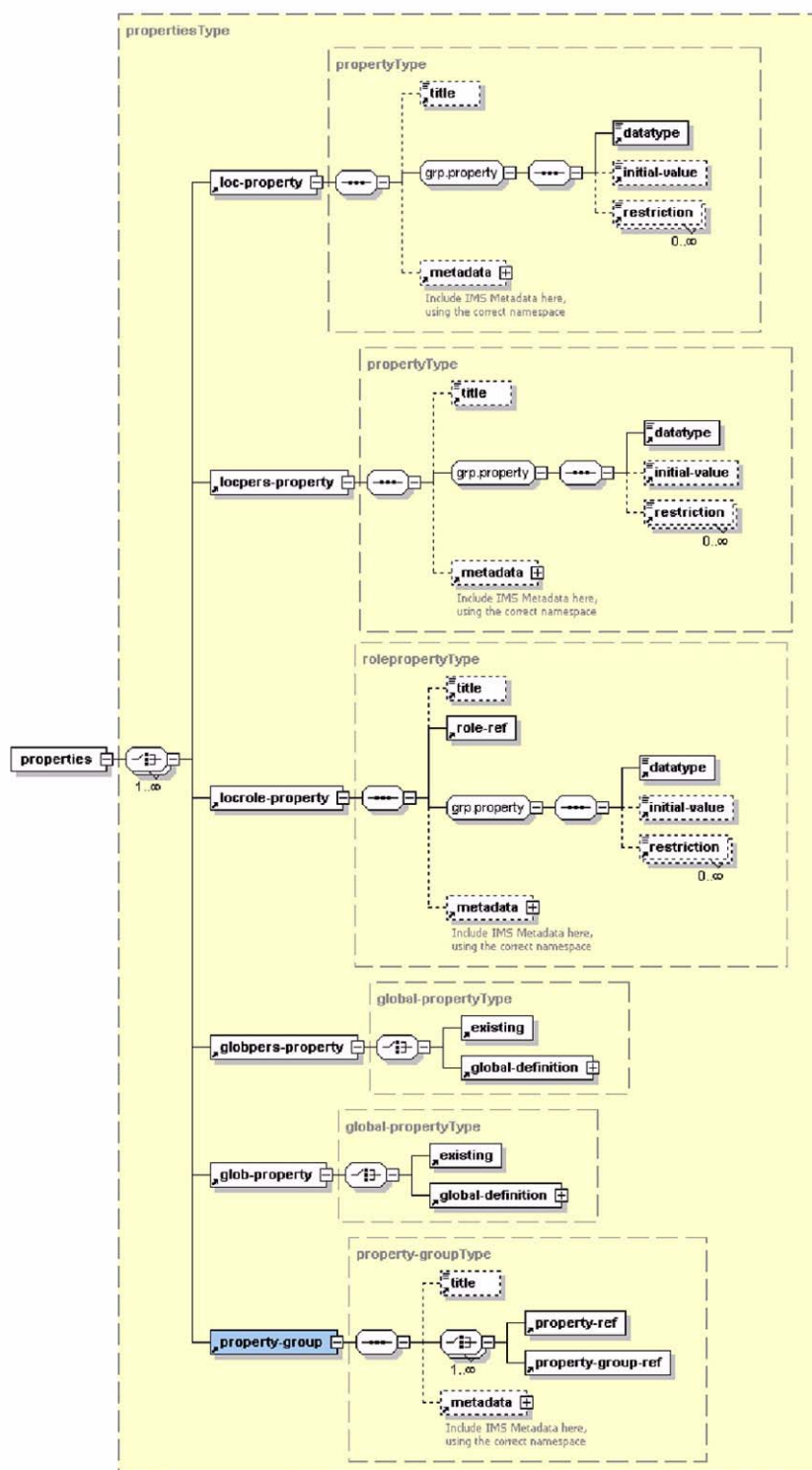
4.12.5 <metadata>

See <metadata> above.

5. Narrative Description of Learning Design Schema Level B

5.1 <properties> Elements

The <properties> element is added to the content model of the <components> element.



Description: Definition and declaration of new and existing properties. All properties referred to in the learning-design are declared in this section, including referenced global-properties that may be defined elsewhere. However, global-properties that are not declared, nor referenced in the learning-design, can be set or viewed with global-elements available in the content resources with type 'imsldcontent'. All properties can be addressed for property-operations (property-ref, view-property, view-property-group, etc.).

Multiplicity: Multiplicity is undetermined at this level. This is true for top level elements of which the multiplicity will be determined in the context of use.

Elements:

- loc-property
- locpers-property
- locrole-property
- globpers-property
- glob-property
- property-group

5.1.1 <loc-property>

Description: Local property, alias: run-property. This property has the same value in a run for every user. The property is owned by the run of the unit-of-learning. The identifier can be used to refer to the property in this unit-of-learning-package. Property-operations can refer to this identifier to operate on the value.

Multiplicity: The <loc-property> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- datatype
- initial-value
- restriction
- metadata

Example:

```
<imsld:loc-property identifier="P-availability-practice-test">
  <imsld:datatype datatype="boolean"/>
</imsld:loc-property>
```

5.1.1.1 <title>

Description: See IMS CP. A short name given to the resource, suitable for rendering in user-agents.

Multiplicity: The <title> element is optional and occurs zero or 1 time.

5.1.1.2 <datatype>

This is where the datatype is provided. The value is set with the attribute containing a fixed set of predefined datatypes. These are also predefined in the attribute datatype. For extensions use the 'other' value and specify the content in the element self.

Multiplicity: The <datatype> element is mandatory and occurs only 1 time.

Attributes:

- **datatype (mandatory).** The fixed set of datatypes to select from. The fixed values to choose from are: boolean, integer, real, string, datetime, duration, text, file, uri and other. The format is described elsewhere in the information model. Data type = token group.

5.1.1.3 <initial-value>

Description: The initial value of the property is set to the value of this element when specified. When this value is not specified the initial value is '<no value>'

Multiplicity: The <initial-value> element is optional and occurs zero or 1 time.

5.1.1.4 <restriction>

Description: Zero or more restrictions of different type may be set on the property-values, meaning that the property-value is valid when it is of the specified data type and its value is within the specified restriction rules. Zero or more restrictions can be specified (these have the same format as specified in the W3C XML schema 1.0 specification) in the attribute: 'restriction-type'. However properties may not contain arrays (lists) of data, but can only contain a single value. So restrictions only apply to this single value. (Also the 'whitespace' restriction is not supported in imsl). The restriction types are specified elsewhere in the information model.

Multiplicity: The <restriction> element is optional and occurs zero or more times.

Attributes:

- **restriction-type (optional).** The fixed set of restriction types supported: minExclusive, minInclusive, maxExclusive, maxInclusive, totalDigits, fractionDigits, length, minLength, maxLength, enumeration, whiteSpace, pattern (see elsewhere in the information model for an explanation). Data type = token group.

5.1.1.5 <metadata>

Description: Include IMS or LOM Meta-data here, using its namespace.

Multiplicity: The <metadata> element is optional and occurs zero or 1 time.

5.1.2 <locpers-property>

Description: Local personal property. This property can have a different value for every user in all the roles for a run of this unit-of-learning. The property is owned by the run of the unit-of-learning, specifying a value per user. The identifier can be used to refer to the property in this unit-of-learning-package. Property operations can refer to this identifier to operate on the value.

Multiplicity: The <locpers-property> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- datatype
- initial-value
- restriction
- metadata

Example:

```
<imsld:locpers-property identifier="P-calculate-advising">
  <imsld:datatype datatype="boolean"/>
</imsld:locpers-property>
```

5.1.2.1 <title>

See <title> above.

5.1.2.2 <datatype>

See <datatype> above.

5.1.2.3 <initial-value>

See <initial-value> above.

5.1.2.4 <restriction>

See <restriction> above.

5.1.2.5 <metadata>

See <metadata> above.

5.1.3 <locrole-property>

Description: Alias: group-property. Local role property. This property has the same value for every user in the specified role during the run of a unit-of-learning. The property is owned by the role in the run of the unit-of-learning. The identifier can be used to refer to the property in this unit-of-learning-package. Property operations can refer to this identifier to operate on the value.

Multiplicity: The <locrole-property> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- role-ref
- datatype
- initial-value
- restriction
- metadata

Example:

```
<imsld:locrole-property identifier="P-Clarify-And-Question">
  <imsld:role-ref ref="R-DD"/>
  <imsld:datatype datatype="boolean"/>
</imsld:locrole-property>
```

5.1.3.1 <title>

See <title> above.

5.1.3.2 <role-ref>

expression.

Multiplicity: The <role-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.1.3.3 <datatype>

See <datatype> above.

5.1.3.4 <initial-value>

See <initial-value> above.

5.1.3.5 <restriction>

See <restriction> above.

5.1.3.6 <metadata>

See <metadata> above.

5.1.4 <globpers-property>

Description: Global personal property, alias: portfolio-property. This property can have a different value for every user, independent of the different runs of units of learning (it specifies the portfolio of the user). The person owns the property. The identifier can be used to refer to the property in this unit-of-learning-package. Property operations can refer to this identifier to operate on the value.

Multiplicity: The <globpers-property> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- existing
- global-definition

Example:

```
<imsld:globpers-property identifier="P-advising-report">  
  <imsld:existing href="" />  
</imsld:globpers-property>
```

5.1.4.1 <existing>

Refers to a property already declared (e.g., in another unit-of-learning, or in the global dossier) to the knowledge of the author (see 'global-definition' what happens if the author defines a new global property which in practice already exists). The property is referred to with href, specifying an absolute URI. NB: when validating this unit-of-learning, the URI does not have to be present. The declaration of the URI by an external unit-of-learning can happen at any time. So this is only under the control of the author.

Multiplicity: The <existing> element is mandatory and occurs only 1 time.

Attributes:

- **href (mandatory).** Refers to a URI. Data type = string.

5.1.4.2 <global-definition>

Description: The global definition can be used to declare and define global properties. Global properties can be defined once and can never be changed from a declaration within the context of a learning design (only outside in the database where the properties reside). Global properties are to be defined with an external definition mechanism preferably. To force consistency, the next rule applies: Once a global property has been defined in whatever context, it

can never be changed! This is also true for re-publications of the same unit-of-learning. So the definition is only used when the URI (href) does not exist yet. Otherwise it is ignored. The URI must be an identifier which identifies the global property global unique. It must be an absolute URI. When the URI is an URL, the URI does not need to point to the property location, but can be interpreted as an identifier.

Multiplicity: The <global-definition> is mandatory and occurs only 1 time.

Attributes:

- **uri (mandatory).** Refers to a URI. Data type = string.

Elements:

- title
- datatype
- initial-value
- restriction
- metadata

5.1.4.2.1 <title>

See <title> above.

5.1.4.2.2 <datatype>

See <datatype> above.

5.1.4.2.3 <initial-value>

See <initial-value> above.

5.1.4.2.4 <restriction>

See <restriction> above.

5.1.4.2.5 <metadata>

See <metadata> above.

5.1.5 <glob-property>

Description: A global property is a global unique property, which stores one value, independent of user, units-of-learning and role. The identifier can be used to refer to the property in this unit-of-learning-package. Property operations can refer to this identifier to operate on the value.

Multiplicity: The <glob-property> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- existing
- global-definition

5.1.5.1 <existing>

See <existing> above.

5.1.5.2 <global-definition>

See <global-definition> above.

5.1.6 <property-group>

Description: A definition of a group of properties that belong together (and are edited in e.g. a form). It can only contain properties of the same type. The identifier can be used to refer to the property-group in this unit-of-learning-package. Operations can refer to this identifier to operate on the value.

Multiplicity: The <property-group> element is mandatory and occurs only 1 time.

Attributes:

- **identifier (mandatory).** An identifier that is unique within the learning design file (ID). Data type = ID.

Elements:

- title
- property-ref
- property-group-ref
- metadata

5.1.6.1 <title>

See <title> above.

5.1.6.2 <property-ref>

Description: Refers to a property This can be a property of all kind: local property, global property, local personal property, local role property, global personal property, local role property. The ref refers to the property declaration in the learning-design. The element can be used as an operand in a calculation or expression.

Multiplicity: The <property-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.1.6.3 <property-group-ref>

Description: Refers to a property-group.

Multiplicity: The <property-group-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.1.6.4 <metadata>

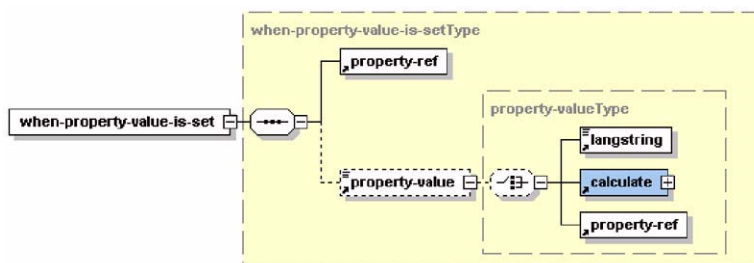
See <metadata> above.

5.2 <when-property-value-is-set> Elements

The <when-property-value-is-set> element is added to the content models as the last element in the group of the following level A Elements:

- <complete-activity>
- <complete-act>

- <complete-play>
- <complete-unit-of-learning>



Description: Simple expression, containing two child elements: a property and an optional property-value. The condition evaluates to true when: 1) the property is set to the specified property-value; 2) the property is not NULL and the property-value is omitted.

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- property-ref
- property-value

Example:

```
<imsld:when-property-value-is-set>
  <imsld:property-ref ref="P-Share-Illustrations"/>
  <imsld:property-value>YES</imsld:property-value>
</imsld:when-property-value-is-set>
```

5.2.1 <property-ref>

See <property-ref> above.

5.2.2 <property-value>

Description: The element can be used as an operand in a calculation or expression. This element specifies the value a property is set or compared to. This depends on the context. For instance within an If statement the property is compared to the value. In a change-property-value context, the property is set to this value. Depending on the property-type this value is of type PCDATA or langstring. Property values may be calculated from the values of other properties. It is also possible to take over the property value of another property (with property-ref).

Multiplicity: The <property-value> element is optional and occurs zero or 1 time.

Elements:

- langstring
- calculate
- property-ref

5.2.2.1 <langstring>

Description: This is identical to the XHTML <p> element. The binding comes from the IMS Meta-Data. The attribute xml:lang may be added to all elements according to the W3C specifications. It is specifically needed on this element, but deleted here in order to support XML schema notation.

Multiplicity: The <langstring> element is mandatory and occurs only 1 time.

5.2.2.2 <calculate>

Description: This is the container for the elements to perform calculations. This container is also used in expressions.

Multiplicity: The <calculate> element is mandatory and occurs only 1 time.

Elements:

- expression

5.2.2.2.1 <expression>

Description: A schema group.

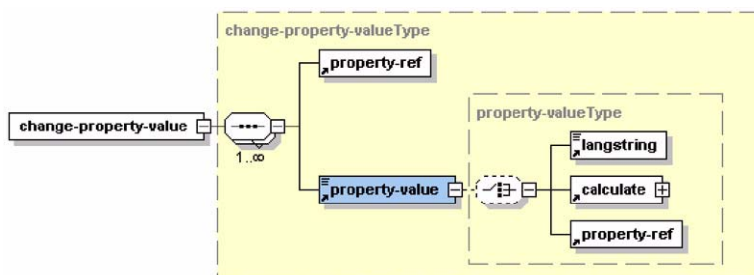
Multiplicity: The <expression> element is mandatory and occurs only 1 time.

5.2.2.3 <property-ref>

See <property-ref> above.

5.3 <change-property-value> Elements

The <change-property-value> element is added to the content model of the level A <on-completion> element. It also occurs in the level B element instance.



Description: This element is used to change values of properties after an event (e.g. completion of something). For example, when the activity is completed, a property value may be changed to reflect this fact. In the dossier also an automated record of completed activities is kept, so it is not necessary to record the completion as such, but to register (or change) other things.

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- property-ref
- property-value

5.3.1 <property-ref>

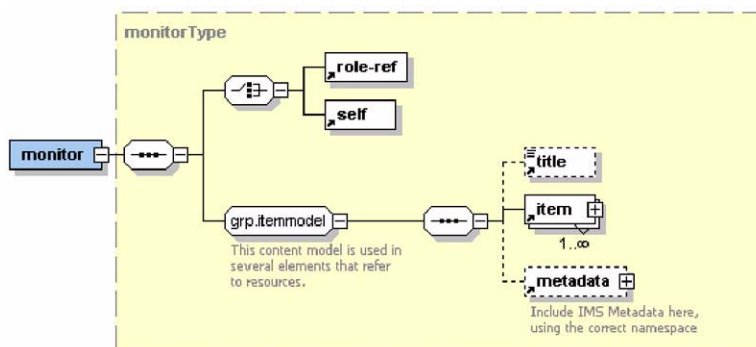
See <property-ref> above.

5.3.2 <property-value>

See <property-value> above.

5.4 <monitor> Elements

The <monitor> element is added to the content model of the level A <service> element.



Description: The monitor service provides a facility for users to look at their own properties or that of others in a structured way. A monitor uses global properties in resources of type 'imsldcontent' to view the properties of one-self or of all users in a role.

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- role-ref
- self
- title
- item
- metadata

5.4.1 <role-ref>

Description: Role refers to the identifier of the resource of the role. The element can be used as an operand in an expression.

Multiplicity: The <role-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.4.2 <self>

Description: Refers to the properties of the person him/her self instead of the role the person is in.

Multiplicity: The <self> element is mandatory and occurs only 1 time.

5.4.3 <title>

See <title> above.

5.4.4 <item>

See <item> above.

5.4.5 <metadata>

See <metadata> above.

5.5 <email-data> Additional Attributes

The <email-data> element has two additional attributes in level B. In level A the element does not have attributes. The attributes are listed below.

Description: This is used for send-mail purposes (as a service in the environment, or in notifications). In level B, the properties on this element refer to the property resources where the relevant e-mail data can be found for the connected role. In level A, the source is not specified explicitly and is left to implementers to decide how to address the data needed. Both properties (email, username) should be available for all persons assigned to the role and the sending party.

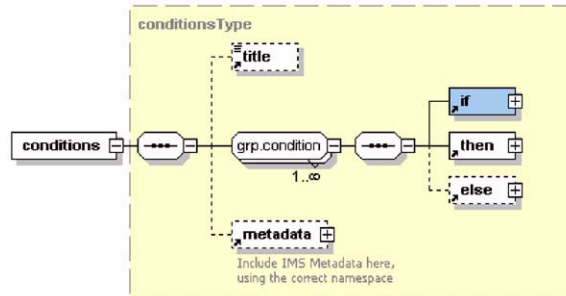
Multiplicity: The <email-data> element is mandatory and occurs 1 or more times.

Attributes:

- **email-property-ref.** Attribute contains a reference to the property containing the email address of the users being notified. Data type = idref.
- **username-property-ref (optional).** This attribute contains a reference to the property containing the user name of the users being notified. Data type = idref.

5.6 <condition> Elements

The <conditions> element is added to the content model of the level A <method> element.



Description: All conditions are pre-conditions and must be evaluated: - when entering the run of a unit of learning (new session); - every time when the value of a property has been changed. This applies only to the following properties: a) properties where the person has access to in the context of the run of the unit of learning, and b) the property has to be evaluated in one of the expressions in the unit of learning. These properties include properties, which are available in the expression, but are set automatically (e.g. time-unit-of-learning-started). An action is performed (fired) according to the success (true) or failure (false) of the condition. The action is to show, hide, change-property-value (level B) or notify (level C) a role. The show and hide actions set the visibility attribute (isvisible) of different objects: activities, environments, items, plays, activity-structures, units-of-learning and different classes of objects (set with the 'class' attribute).

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- title
- if

- then
- else
- metadata

Example:

```

<imsld:conditions>
  <!--the condition below hides or shows examples in the background document that the students
may use-->
  <imsld:if>
    <imsld:is>
      <imsld:property-ref ref="P-availability-examples"/>
      <imsld:property-value>with examples</imsld:property-value>
    </imsld:is>
  </imsld:if>
  <imsld:then>
    <imsld:show>
      <imsld:class class="C-examples"/>
    </imsld:show>
  </imsld:then>
  <!--the condition below shows the practice and testing activity structure and the background
environment to the student once the preparations have been completed-->
  <imsld:if>
    <imsld:not>
      <imsld:or>
        <imsld:no-value>
          <imsld:property-ref ref="P-completion-test-advising"/>
        </imsld:no-value>
        <imsld:no-value>
          <imsld:property-ref ref="P-completion-test-anticipating"/>
        </imsld:no-value>
      </imsld:or>
    </imsld:not>
  </imsld:if>
  <imsld:then>
    <imsld:show>
      <imsld:learning-activity-ref ref="LA-request-grade"/>
      <imsld:environment-ref ref="E-background"/>
    </imsld:show>
  </imsld:then>
  <!--the condition below shows the request test activity to the student once the practice and
testing activity structure has been completed-->
  <imsld:if>
    <imsld:is>
      <imsld:property-ref ref="P-availability-practice-test"/>
      <imsld:property-value>true</imsld:property-value>
    </imsld:is>
  </imsld:if>
  <imsld:then>
    <imsld:show>
      <imsld:activity-structure-ref ref="AS-practice-and-testing"/>
    </imsld:show>
  </imsld:then>
</imsld:conditions>

```

5.6.1 <title>

See <title> above.

5.6.2 <if>

Description: If refers to an expression-schema which evaluation results in the value: true or false. Different expression schemas may be used, however it is preferred to use the expression (and calculation) schema provided with IMSLD. When the expression resolves to 'true', the 'then' rule fires. When it resolves to 'false' the 'else' rule fires when it is present (otherwise nothing happens in this rule).

Multiplicity: The <if> element is mandatory and occurs only 1 time.

Elements:

- {expression} schema group, see below.

Example:

```
<imsld:if>
  <imsld:is>
    <imsld:property-ref ref="P-availability-examples"/>
    <imsld:property-value>with examples</imsld:property-value>
  </imsld:is>
</imsld:if>
<imsld:then>
  <imsld:show>
    <imsld:class class="C-examples"/>
  </imsld:show>
</imsld:then>
```

5.6.3 <then>

Description: When expression specified in the <if> element is true, the statements after the <then> element are executed.

Multiplicity: The <then> element is mandatory and occurs only 1 time.

Elements:

- {thenmodel} schema group, see below.

Example:

```
<imsld:if>
  <imsld:is>
    <imsld:property-ref ref="P-availability-examples"/>
    <imsld:property-value>with examples</imsld:property-value>
  </imsld:is>
</imsld:if>
<imsld:then>
  <imsld:show>
    <imsld:class class="C-examples"/>
  </imsld:show>
</imsld:then>
```

5.6.4 <else>

Description: Is executed when the If expression is false.

Multiplicity: The <else> element is optional and occurs zero or 1 time.

Elements:

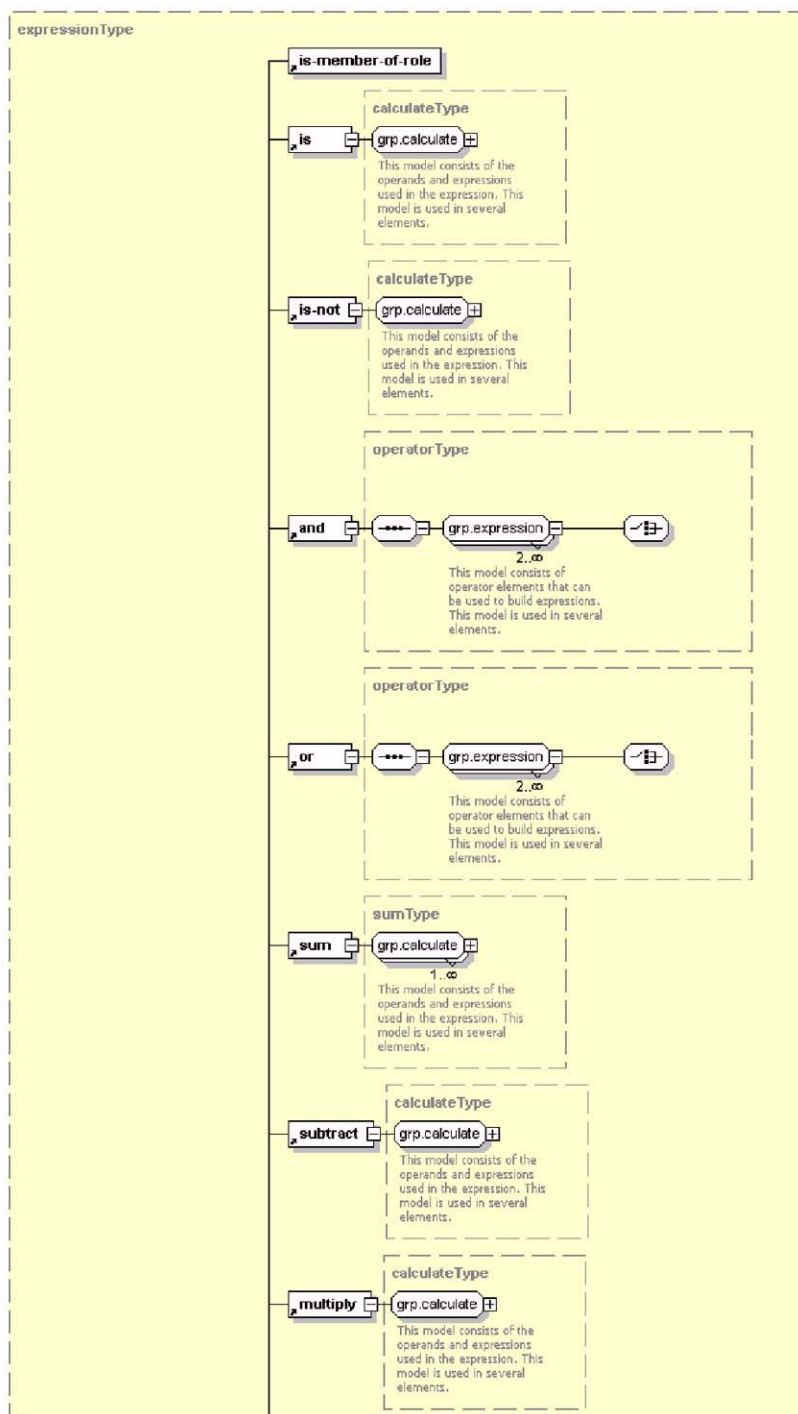
- {thenmodel} schema group, see below.

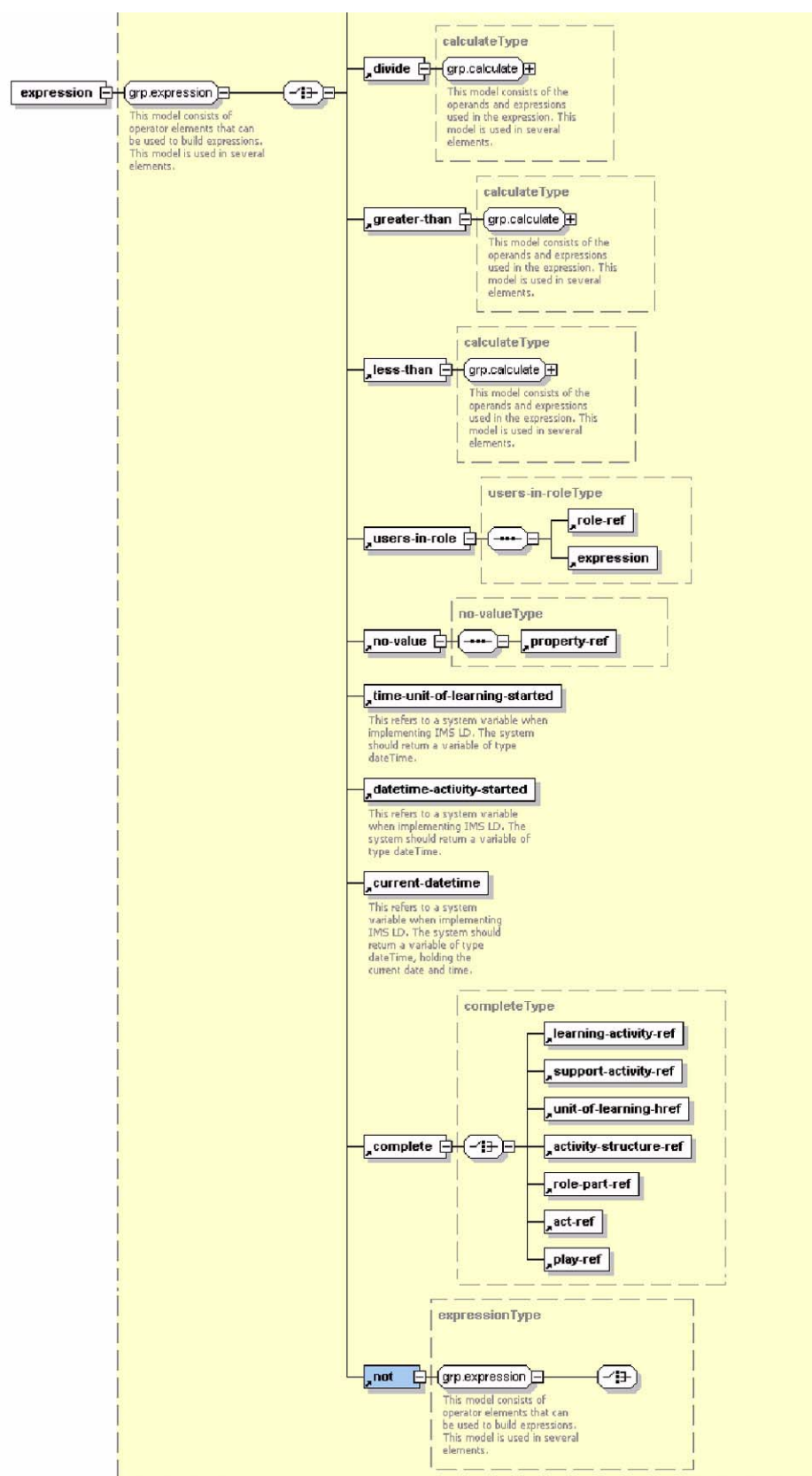
5.6.5 <metadata>

See <metadata> above.

5.7 {expression} Schema Group

The model for *if* contains two grouping entities named *expression* and *calculate*. The entity *expression* is also available as an element with the same content in the model of element *when-condition-true* (see later). The entity *calculate* exists as a type with the identical structure within the content model of the element *property-value* (see: *change-property-value*). To avoid repeating the same information several times, the models are only expressed here.





5.7.1 <is-member-of-role>

Description: Is true when the person is a member of the role referenced with 'ref'.

Multiplicity: The <is-member-of-role> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.2 <is>

Description: Regular logical operator (true when an expressions is true).

Multiplicity: The <is> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

Example:

```
<imsld:if>
  <imsld:is>
    <imsld:property-ref ref="P-availability-examples"/>
    <imsld:property-value>with examples</imsld:property-value>
  </imsld:is>
</imsld:if>
<imsld:then>
  <imsld:show>
    <imsld:class class="C-examples"/>
  </imsld:show>
</imsld:then>
```

5.7.3 <is-not>

Description: Complement of regular logical operator (true when an expressions is false).

Multiplicity: The <is-not> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.4 <and>

Description: Regular logical operator (true when both expressions are true).

Multiplicity: The <and> element is mandatory and occurs only 1 time.

Elements:

- {expression} schema group, see above.
- {expression} schema group, see above.

Example:

```
<imsld:and>
  <imsld:no-value>
    <imsld:property-ref ref="P-completion-test-advising"/>
  </imsld:no-value>
  <imsld:no-value>
    <imsld:property-ref ref="P-completion-test-anticipating"/>
  </imsld:no-value>
```

</imsld:and>

5.7.5 <or>

Description: Regular logical operator (true when at least one of both expressions is true).

Multiplicity: The <or> element is mandatory and occurs only 1 time.

Elements:

- {expression} schema group, see above.
- {expression} schema group, see above.

Example:

```
<imsld:or>
  <imsld:no-value>
    <imsld:property-ref ref="P-completion-test-advising"/>
  </imsld:no-value>
  <imsld:no-value>
    <imsld:property-ref ref="P-completion-test-anticipating"/>
  </imsld:no-value>
</imsld:or>
```

5.7.6 <sum>

Description: Regular calculation operator (returns the sum of the values provided).

Multiplicity: The <sum> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.7 <subtract>

Description: Regular calculation operator (returns the result of the first value minus the second value provided).

Multiplicity: The <subtract> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.8 <multiply>

Description: Regular calculation operator (returns the sum of the values provided).

Multiplicity: The <multiply> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.9 <divide>

Description: Regular calculation operator (returns the result of the division of the first value by the second value).

Multiplicity: The <divide> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.10 <greater-than>

Description: Regular logical operator (is true when the first value is greater than the second value).

Multiplicity: The <greater-than> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.11 <less-than>

Description: Regular logical operator (is true when the first value is less than the second value).

Multiplicity: The <less-than> element is mandatory and occurs only 1 time.

Elements:

- {calculate} schema group, see below.

5.7.12 <users-in-role>

Description: Contains a sequence of elements that specify that the expression applies to all the individual members of a referenced role (and not to the role itself).

Multiplicity: The <users-in-role> element is mandatory and occurs only 1 time.

Elements:

- role-ref
- {expression} schema group, see above.

5.7.12.1 <role-ref>

See <role-ref> above.

5.7.13 <no-value>

Description: True when a property is empty.

Multiplicity: The <no-value> element is mandatory and occurs only 1 time.

Elements:

- property-ref

5.7.14 <time-unit-of-learning-started>

Refers to the time that the run of the unit of learning has been started. This is a fixed time, stored during the instantiation of the learning design.

Multiplicity: The <time-unit-of-learning-started> element is mandatory and occurs only 1 time.

Attributes:

- **unit-or-learning-uri (mandatory).** Data type = string.

5.7.15 <daytime-activity-started>

Description: Date and time when an activity has been first accessed by an individual user. Uses the datetime format (see datatype).

Multiplicity: The <datetime-activity-started> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.16 <current-datetime>

Description: Current date and time. Uses the datetime format (see datatype).

Multiplicity: The <current-datetime> element is mandatory and occurs only 1 time.

5.7.17 <complete>

Description: Is true when the related element (e.g. activity) has been completed.

Multiplicity: The <complete> element is mandatory and occurs only 1 time.

Elements:

- learning-activity-ref
- support-activity-ref
- unit-of-learning-href
- activity-structure-ref
- role-part-ref
- act-ref
- play-ref

5.7.17.1 <learning-activity-ref>

Description: Refers to a learning-activity. The element can be used as an operand in a calculation or expression.

Multiplicity: The <learning-activity-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.17.2 <support-activity-ref>

Description: Refers to a support-activity. The element can be used as an operand in a calculation or expression.

Multiplicity: The <support-activity-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.17.3 <unit-of-learning-href>

Description: The element can be used as an operand in a calculation or expression. This element refers to the resource of a unit-of-learning (uol). That can be the resource of this current unit-of-learning, the resource of a unit-of-learning present in the package or a resource pointing to a unit-of-learning outside of the package (absolute URI expressed as an ID).

Multiplicity: The <unit-of-learning-href> element is mandatory and occurs only 1 time.

Attributes:

- **href (mandatory).** Refers to a URI. Data type = string.

5.7.17.4 <activity-structure-ref>

Description: Refers to an activity-structure.

Multiplicity: The <activity-structure-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.17.5 <role-part-ref>

Description: Refers to a role-part.

Multiplicity: The <role-part-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.17.6 <act-ref>

Description: Refers to an act (in method/play/act).

Multiplicity: The <act-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.17.7 <play-ref>

Description: Refers to a play.

Multiplicity: The <play-ref> element is mandatory and occurs only 1 time.

Attributes:

- **ref (mandatory).** Refers to an identifier within the learning design. Data type = idref.

5.7.18 <not>

Description: Regular logical expression. Complements the sense of the embedded expression.

Multiplicity: The <not> element is mandatory and occurs only 1 time.

Elements:

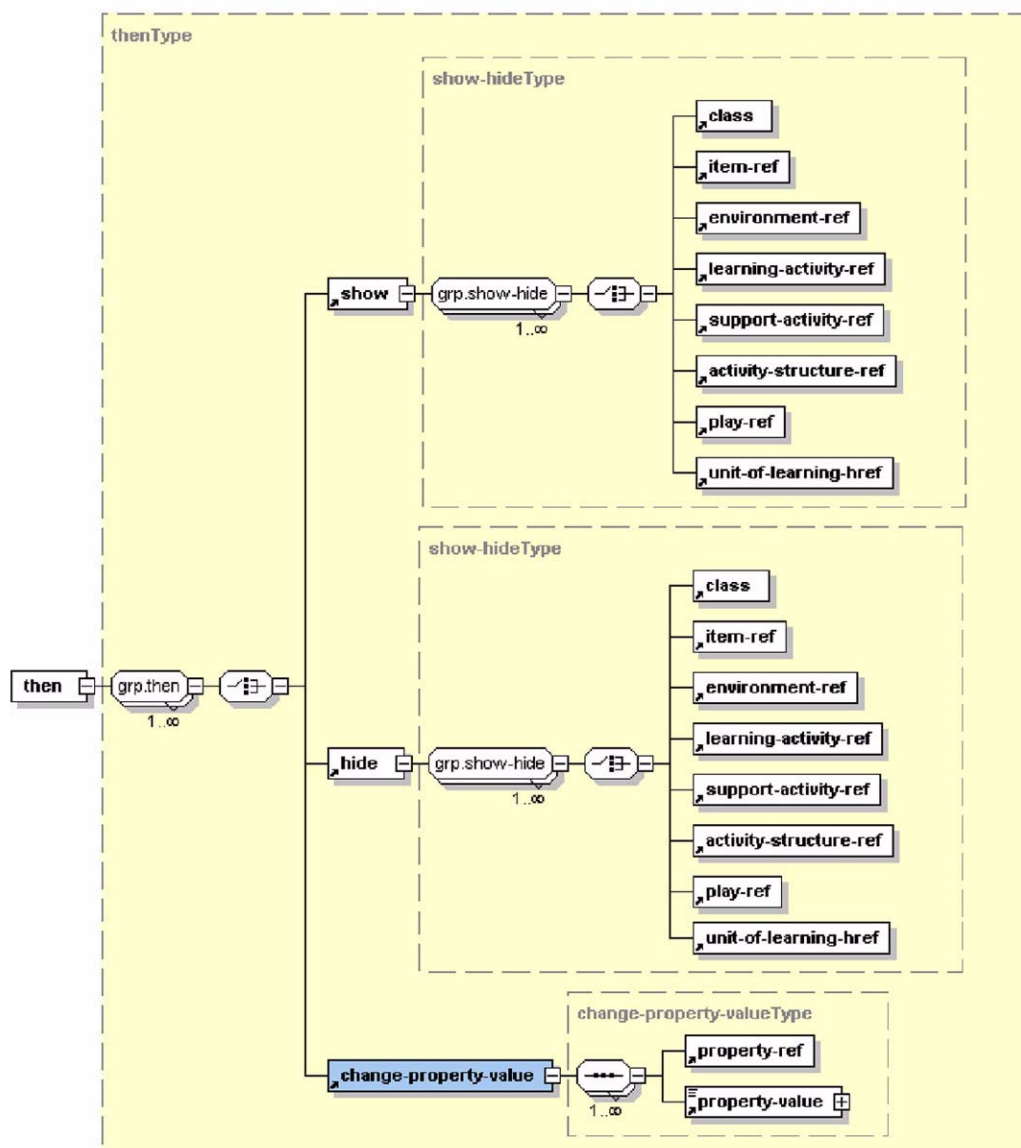
- {expression} schema group, see above.

Example:

```
<imsld:not>
  <imsld:or>
    <imsld:no-value>
      <imsld:property-ref ref="P-completion-test-advising"/>
    </imsld:no-value>
    <imsld:no-value>
      <imsld:property-ref ref="P-completion-test-anticipating"/>
    </imsld:no-value>
  </imsld:or>
</imsld:not>
```


5.8 {thenmodel} Schema Group

The {thenmodel} schema group contains the following elements.



5.8.1 <show>

Description: A choice of elements to specify what has to be shown when the condition (if) is true. This has effect on the 'invisible' status of the entity (set to true).

Multiplicity: The <show> element is mandatory and occurs only 1 time.

Elements:

- class
- item-ref
- environment-ref
- learning-activity-ref
- support-activity-ref

- activity-structure-ref
- play-ref
- unit-of-learning-href

Example:

```
<imsld:if>
  <imsld:is>
    <imsld:property-ref ref="P-availability-examples"/>
    <imsld:property-value>with examples</imsld:property-value>
  </imsld:is>
</imsld:if>
<imsld:then>
  <imsld:show>
    <imsld:class class="C-examples"/>
  </imsld:show>
</imsld:then>
```

5.8.1.1 <class>

Description: Indicates that elements with a certain class attribute value must be shown or hidden, depending on the context of the element (in show or hide). The class attribute is a global attribute and may be set in resources of type ‘imsldcontent’ and is available on the environment elements in the learning-design model. Note that the classes can be used for style sheet like functions (e.g. set visibility), but they can also have a semantic classification purpose (just as in HTML) not connected to style sheets or automated processing at all. Note: this is similar to the EML 1.0 ‘content-type’ attribute. It is used to identify classes of common objects in order to manipulate them once.

Multiplicity: The <class> element is mandatory and occurs only 1 time.

Attributes:

- **class (optional).** The class attribute refers to the value of class attributes available in learning-design or content elements. Contains a CDATA string. Just as in HTML more than one class may be specified in one CDATA string, each separated with a blank space. The priority order for classes is the same as specified in the CSS specification (see <http://www.w3.org/style/css>). Any element can in principle have the class attribute. ‘Class’ is a global defined W3C attribute for html 4.0 and XHTML [LD14]. This attribute assigns a class name or set of class names to an element. Any number of elements may be assigned the same class name or names. Multiple class names must be separated by white space characters. The class element can be used for semantic grouping of elements and be manipulated by IMSLD conditions and stylesheets. When sending a learning object to a web client, include the class attribute and value(s). Data type = string.
- **title (optional).** When the content is collapsed (see ‘with-control’) a title has to be given. The title is provided in the ‘title’ attribute on the class element. Data type = string.
- **with-control (optional).** Boolean: when true the content elements are hidden, but in the user-interface a collapse and expand control (like the [+] controls in windows explorer) is provided. With this control a user can decide to hide or show the content in the element him or her self. Data type = string.

5.8.1.2 <item-ref>

See <item-ref> above.

5.8.1.3 <environment-ref>

See <environment-ref> above.

5.8.1.4 <learning-activity-ref>

See <learning-activity-ref> above.

5.8.1.5 <support-activity-ref>

See <support-activity-ref> above.

5.8.1.6 <activity-structure-ref>

See <activity-structure-ref> above.

5.8.1.7 <play-ref>

See <play-ref> above.

5.8.1.8 <unit-of-learning-href>

See <unit-of-learning-href> above.

5.8.2 <hide>

Description: A choice of elements to specify what must be made hidden when the condition (if) is true. This affects the 'invisible' status of the entity (set to false).

Multiplicity: The <hide> element is mandatory and occurs only 1 time.

Elements:

- class
- item-ref
- environment-ref
- learning-activity-ref
- support-activity-ref
- activity-structure-ref
- play-ref
- unit-of-learning-href

5.8.2.1 <class>

See <class> above.

5.8.2.2 <item-ref>

See <item-ref> above.

5.8.2.3 <environment-ref>

See <environment-ref> above.

5.8.2.4 <learning-activity-ref>

See <learning-activity-ref> above.

5.8.2.5 <support-activity-ref>

See <support-activity-ref> above.

5.8.2.6 <activity-structure-ref>

See <activity-structure-ref> above.

5.8.2.7 <play-ref>

See <play-ref> above.

5.8.2.8 <unit-of-learning-href>

See <unit-of-learning-href> above.

5.8.3 <change-property-value>

Description: This element is used to change values of properties after an event (e.g. completion of something). For example, when the activity is completed, a property value may be changed to reflect this fact. In the dossier also an automated record of completed activities is kept, so it is not necessary to record the completion as such, but to register (or change) other things.

Multiplicity: The <change-property-value> element is mandatory and occurs only 1 time.

5.8.3.1 <proper-ref>

Description: A reference to a property.

Multiplicity: The <proper-ref> element is mandatory and occurs only 1 time.

5.8.3.2 <property-value>

Description: The new property value.

Multiplicity: The <property-value> element is mandatory and occurs only 1 time.

5.8.3.2.1 <langstring>

See <langstring> above.

5.8.3.2.2 <calculate>

See {calculate} Schema Group below.

5.8.3.2.3 <property-ref>

See <property-ref> above.

5.9 <calculate>

Description: Calculate the value of the embedded expressions.

Multiplicity: The <calculate> element is mandatory and may occur multiple times.

Elements:

- {expression} schema group, see above.

Example:

```
<imsld:calculate>
  <imsld:not>
    <imsld:or>
      <imsld:no-value>
        <imsld:property-ref ref="P-advising-assessment"/>
      <imsld:no-value>
        <imsld:property-ref ref="P-advising-testgrade"/>
    </imsld:or>
  </imsld:not>
</imsld:calculate>
```

```
        </imsld:no-value>
      </imsld:or>
    </imsld:not>
  </imsld:calculate>
```

5.10 <when-condition-true>

The <when-condition-true> element is added to the content model of <complete-act>, which was already extended with the element <when-property-value-is-set>.

Description: Simple expression for a condition. This condition applies to all the individual users mentioned in the containing role-ref. When the contained expression is true for all users in the specified roles, this condition is true.

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- role-ref
- expression

5.10.1 <role-ref>

See <role-ref> above.

5.10.2 <expression>

Description: Contains a choice of different expression elements.

Multiplicity: The <expression> element is mandatory and occurs only 1 time.

Elements:

- {expression} schema group, see above.

5.11 <global-elements> Elements

There are four global elements defined in the IMSLD specification:

- view-property
- view-property-group
- set-property
- set-property-group

In level B these elements are all empty. The global elements must each be used separately – that is without the container <global-elements>, within any XML content schema (like XHTML). The <global-elements> elements do not have any other function than temporary grouping the different global elements.

Note: Global elements are not part of the *learning-design* model! Use the standard namespace

Description: Abstract wrapper for a choice of elements, used for temporary grouping of the global elements. It has no other function at all. It never occurs in content or any document instances. When global-elements, like set-property are used, they are always used on their own and not in the context of this wrapper.

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- view-property
- view-property-group
- set-property
- set-property-group

5.11.1 <view-property>

Description: Global element, to be included in an external XML content schema. The resource type is 'imsldcontent'. With this element a specified property-value may be viewed. It works outside of the context of a text line (e.g. outside the context of a <p> element. The view attribute sets whether the value or the title+value should be delivered. It refers to the property with an ref or href. In order to avoid confusions it is good practice to include the imslldcontent with property-operations in the unit-of-learning-package. In case of personal properties, the property value of the user himself is returned. Only in the context of a monitor object (with the role-ref specified) the values of all the users in the specified role(s) are returned.

Multiplicity: The <view-property> element is mandatory and occurs only 1 time.

Attributes:

- **href (optional).** Refers to a URI. Data type = string.
- **property-of (mandatory).** Fixed selection: 'self' or 'supported-person'. When 'self' is selected only the value of the property in the person using the property is referenced. When 'supported-persons' is selected, the properties of all the persons selected are. Data type = token group.
- **ref (optional).** Refers to an identifier within the learning design. Data type = idref.
- **view (mandatory).** Fixed selection: 'value' or 'title-value'. With 'value' only the value is returned for the display of the property. With 'title-value', the title of the property and the value is returned; in a property-group also the title of the group is returned. Data type = token group.

5.11.2 <view-property-group>

Description: Global element, to be included in an external XML content schema. The resource type is 'imsldcontent'. With this element the values of the properties in a specified property-group may be viewed. It works outside of the context of a textline (e.g. outside <p>). The view attribute determines whether the titles of the containing values should be shown. The group-title is always shown. It refers to the property-group identifier or URI with a ref or href. In order to avoid confusions it is good practice to include the imslldcontent with property-operations in the unit-of-learning-package. In case of personal properties, the property value of the user himself is returned. Only in the context of a monitor object (with the role-ref specified) the values of all the users in the specified role(s) are returned.

Multiplicity: The <view-property-group> element is mandatory and occurs only 1 time.

Attributes:

- **href (optional).** Refers to a URI. Data type = string.
- **property-of (mandatory).** Fixed selection: 'self' or 'supported-person'. When 'self' is selected only the value of the property in the person using the property is referenced. When 'supported-persons' is selected, the properties of all the persons selected are. Data type = token group.
- **ref (optional).** Refers to an identifier within the learning design. Data type = idref.
- **view (mandatory).** Fixed selection: 'value' or 'title-value'. With 'value' only the value is returned for the display of the property. With 'title-value', the title of the property and the value is returned; in a property-group also the title of the group is returned. Data type = token group.

5.11.3 <set-property>

Description: Global element, to be included in an external XML content schema. The resource type is 'imsldcontent'. With this element a specified property-value may be set by the user. It works outside of the context of a textline (e.g. outside <p>). The view attribute sets whether the value or the title+value should be delivered. The user gets a control

in the user-interface to set the value of the property. The type of control is dependent on the property datatype and the restrictions. In the control the current value is shown and the datatype and restrictions are made explicit so that the user knows exactly what values are valid and which are not. This allows for client-side checking of the input (dependent on implementation this may also be dealt with at the server side). The element refers to the property URI or identifier with a ref or href. In order to avoid confusions it is good practice to include the `imsldcontent` with property-operations in the unit-of-learning-package. In case of personal properties, the property value of the user himself is set. Only in the context of a monitor object (with the role-ref specified) the values of all the users in the specified role can be set.

Multiplicity: The `<set-property>` element is mandatory and occurs only 1 time.

Attributes:

- **href (optional).** Refers to a URI. Data type = string.
- **max-transactions (optional).** The attribute max-transactions represents the number of times a property may set by a user. Technical (upload) errors do not count as a trial, but only successful transactions. When the attribute is not specified, the number of attempts is set to unlimited. Data type = string.
- **property-of (mandatory).** Fixed selection: 'self' or 'supported-person'. When 'self' is selected only the value of the property in the person using the property is referenced. When 'supported-persons' is selected, the properties of all the persons selected are. Data type = token group.
- **ref (optional).** Refers to an identifier within the learning design. Data type = idref.
- **transaction-type (optional).** The attribute transaction-type: future extension, e.g. for secure transactions. Data type = string
- **view (mandatory).** Fixed selection: 'value' or 'title-value'. With 'value' only the value is returned for the display of the property. With 'title-value', the title of the property and the value is returned; in a property-group also the title of the group is returned. Data type = token group.

5.11.4 `<set-property-group>`

Description: Global element, to be included in an external XML content schema. The resource type is 'imsldcontent'. With this element the values of the properties contained in a specified property-group may be set by the user. It works outside of the context of a text line (e.g. outside `<p>`). The view attribute determines whether the titles of the containing values should be shown. The group-title is always shown. The user gets a control in the user-interface to set the value of the property-group. The type of control per property is dependent on the property datatype and the restrictions set on the property. In the control the current value of the properties are shown and the data type and restrictions are made explicit so that the user knows exactly what values are valid and which are not. This allows for client-side checking of the input (dependent on implementation this may also be dealt with at the server side). All values of all properties in the group are set at by the user before updating. The transaction is always counted for the group of properties, not for single properties. The element refers to the property-group identifier or URI with a ref or href. In order to avoid confusions it is good practice to include the `imsldcontent` with property-operations in the unit-of-learning package. In case of personal properties, the property values of the user himself is set. Only in the context of a monitor object (with the role-ref specified) the values of all the users in the specified role can be set.

Multiplicity: The `<set-property-group>` element is mandatory and occurs only 1 time.

Attributes:

- **href (optional).** Refers to a URI. Data type = string.
- **max-transactions (optional).** The attribute max-transactions represents the number of times a property may set by a user. Technical (upload) errors do not count as a trial, but only successful transactions. When the attribute is not specified, the number of attempts is set to unlimited. Data type = string.
- **property-of (mandatory).** Fixed selection: 'self' or 'supported-person'. When 'self' is selected only the value of the property in the person using the property is referenced. When 'supported-persons' is selected, the properties of all the persons selected are. Data type = token group.
- **ref (optional).** Refers to an identifier within the learning design. Data type = idref.

- **transaction-type (optional).** The attribute transaction-type: future extension, e.g. for secure transactions. Data type = string
- **view (mandatory).** Fixed selection: 'value' or 'title-value'. With 'value' only the value is returned for the display of the property. With 'title-value', the title of the property and the value is returned; in a property-group also the title of the group is returned. Data type = token group.

5.12 Global Attribute 'class'

Conditions can show or hide elements with the attribute *class*. This attribute is a global attribute – defined by the W3C in the context of Cascading Style Sheets (CSS) - that is available at the following elements within the learning-design model:

- learning-object
- service

Outside the context of learning design it can be added to any XML content schema. It is available on all elements in XHTML. Conditions do not only effect the class attributes within the learning design, but also in the content, when this content is of resource type 'imsldcontent'.

5.13 Data Types

In the property declaration the following datatypes are used. The format per datatype is specified:

- *Boolean*: Represents binary logic, true or false (aliases: yes/no; 1/0).
NB: just as any other data types, booleans can also have <no-value>.
- *Integer*: Represents the standard mathematical concept of integer numbers, representing whole positive and negative numbers (including zero), ranging from:
-9223372036854775898 to 9223372036854775807 (alias: longinteger).
- *Real*: Represents the standard mathematical concept representing arbitrary precision decimal numbers, and must be capable of handling a number to 18 decimal places at least.
- *String*: Represents any legal character strings. The minimal maximum number of characters is 2000.
- *File*: Represents any binary file as datatype. The property stores this file.
- *Uri*: Represents an URI according to the IETF's RFC 2396
Note: according to the w3c only the word URI should be used in future and not URL or URN. (see: <http://www.w3.org/TR/2001/NOTE-uri-clarification-20>)
- *Datetime*: Specifies the date and time in the form: CCYY-MM-DDThh:mm:ss
CC is the century; YY is the year (year 0000 is prohibited); MM is the month; dd is the day.
T is the date/time separator. hh are the hours; mm are the minutes; ss are the seconds.
(see ISO 8601). There is also an optional timezone separator. Partial productions of the lexical expression are not allowed.
- *Duration*: Specifies an amount of time: the duration of an event in relative terms (e.g. the duration given the start datetime of the run of a unit-of-learning. The format - also used in the W3C XML schema specification - is:
PnYnMnDnHnMnS where:
P is the designator that must always be present.
n is a variable where an integer is filled in.
nY represents the number of years
nM represents the number of month
nD represents the number of days
T is the date/time separator which must always be present when representing time.
nH is the number of hours

nM is the number of minutes

nS is the number of seconds.

Example: P2Y0M1DT20H10M55S, Meaning that the duration is: 2 years and 0 month and 1 day and 20 hours and 10 minutes and 55 seconds. Limited forms of lexical production are also allowed: E.g. a duration of 40 minutes is expressed: PT40M. A duration of 30 days is: P30D text: represents any legal character strings. The minimal maximum number of characters is 64000 (about 10 pages of A4 text).

5.14 Restriction Types

For properties certain restrictions on the data values can be specified. The restriction types are:

- *length*: constraints the length of the property value of a textual datatype (string, text or uri) in terms of the of characters that it can have.
- *minLength*: constraints the minimum number of characters that a property of textual datatype can have for its value.
- *maxLength*: constraints the maximum number of units of length that a property of textual datatype can have for its value.
- *enumeration*: constraints the value of a property to a specific value (use for value alternative lists).
- *maxInclusive*: constraints the value of an ordered (integer, real, datetime) property to a specific inclusive upper bound.
- *minInclusive*: constraints the value of a ordered property to a specific inclusive lower bound.
- *maxExclusive*: constraints the value of a ordered property to a specific exclusive upper bound.
- *minExclusive*: constraints the value of a ordered property to a specific exclusive lower bound.
- *totalDigits*: constraints the value of a decimal property to a specific number of digits it must contain.
- *fractionDigits*: constraints the value of a decimal property to the maximum number of digits it may have after the decimal point.*pattern*: constraints the literals comprising the value of a property to a pattern defined by a regular expression.

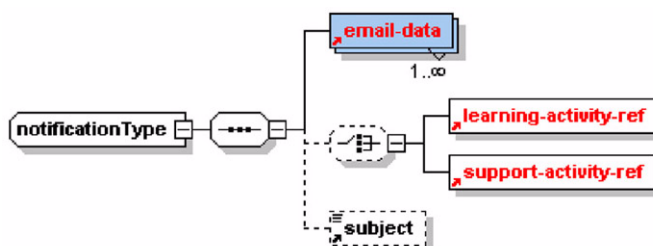
6. Narrative Description of Learning Design Schema Level C

Level C adds the capability for a learning designer to specify the sending of messages and setting of new activities based on certain events. The runtime system, or ‘user-agent’ is expected to support a notification mechanism. Notifications are event driven mechanisms, which can be directed towards elements in the system or to human users.

Notifications effects the following content models of level B elements:

- The *on-completion* model is extended with a <notification> element.
- The *then* model is extended with a <notification> element.
- Global elements: *set-property* and *set-property-group* are both extended with a <notification> element.

6.1 <notification> Elements



Description: A notification happens after an event that is known by the runtime environment. Such an event can be e.g.: the completion of an activity, an expression evaluates to true, or a property value is set. The notification makes a new learning activity or a new support activity active for a role or it only sends a message. A notification is of the highest priority, meaning that an otherwise invisible item will be made visible and accessible to the user. Depending on the implementation an email message can be send to the user, notifying that a new activity has arrived (with a link to that activity in the message). For the mail header the subject field can be filled in to a specific value (otherwise a standard message will be send). A notification can be inserted in external vocabularies (after an event like set-property), however, then the content must be provided in the package (because it contains references to identifiers in the package). When the identifier cannot be resolved the notification is ignored (but does not stop the xhtml content to be presented).

Multiplicity: Multiplicity is undetermined at this level. This is true for top-level elements of which the multiplicity will be determined in the context of use.

Elements:

- email-data
- learning-activity-ref
- support-activity-ref
- subject

Example:

```
<imsld:notification>
  <imsld:email-data email-property-ref="P-email">
    <imsld:role-ref ref="R-tutor"/>
  </imsld:email-data>
  <imsld:subject>Test advising report submitted</imsld:subject>
</imsld:notification>
```

6.1.1 <email-data>

Description: This is used for send-mail purposes (as a service in the environment, or in notifications). In Level B, the properties on this element refer to the property resources where the relevant e-mail data can be found for the connected role. In Level A, the source is not specified explicitly and is left to implementers to decide how to address the data needed. Both properties (email, username) should be available for all persons assigned to the role and the sending party.

Multiplicity: The <email-data> element is mandatory and occurs 1 or more times.

Attributes:

- **email-property-ref.** Attribute contains a reference to the property containing the email address of the users being notified. Data type = idref.
- **username-property-ref (optional).** This attribute contains a reference to the property containing the user name of the users being notified. Data type = idref.

Elements:

- role-ref

6.1.1.1 <role-ref>

See <role-ref> above.

6.1.2 <learning-activity-ref>

See <learning-activity-ref> above.

6.1.3 <support-activity-ref>

See <support-activity-ref> above.

6.1.4 <subject>

Description: Specifies the subject of a notification, to be presented to the notified actor when the notification is activated. For example, in the mail-header (subject field).

Multiplicity: The <subject> element is optional and occurs zero or 1 time.

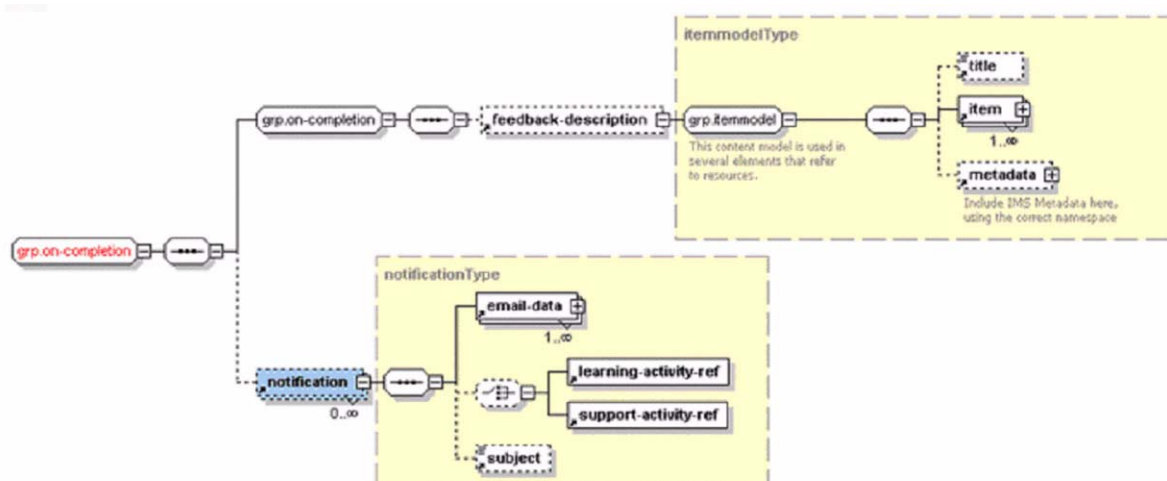
6.2 <notification> Extensions

Notifications are extensible to the following models:

- on-completion
- then
- global-elements

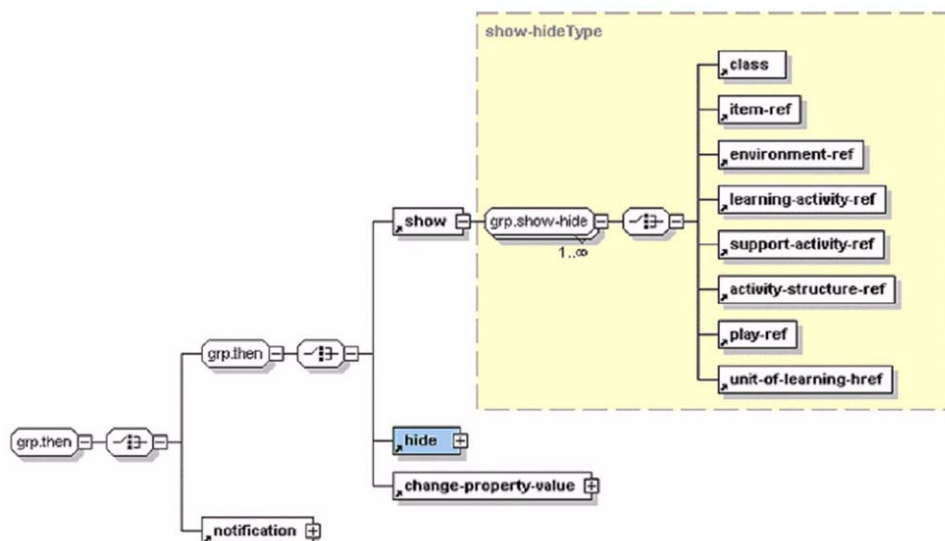
6.2.1 <on-completion>

Notifications are added to the <on-completion> model:



6.2.2 <then>

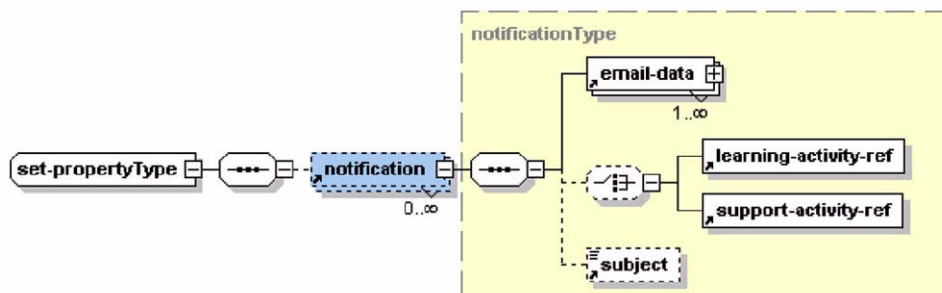
Notifications are added to the <then> model:



6.2.3 <global-elements>

Notifications are added to the following <global-elements>:

- set-property
- set-property-group



7. Normative Points

7.1 XML Schema File Naming Convention

Conformant with IMS file naming conventions, an IMS Learning Design XML Schema binding instance shall be named according to this syntactical model: `imsld_vmpn[pr[ps]].xsd`, where

- “`imskd_`” signifies that the file pertains to the IMS Learning Design Specification,
- “`vm`” is a major version of the IMS Learning Design Specification defined by a binding instance, where “`m`” is the major version number, and is always followed by
- “`pn`”, which signifies a minor version of the IMS Learning Design Specification defined by a binding instance, where “`n`” is minor version number, and where
- `[pr[ps]]` is an optional set of major release and minor release numbers (as indicated by the brackets which are *not* part of the file name), depending on changes to schema instances themselves that are not driven by a change to a normative instance of the IMS Learning Design Specification, with
- “`pr`” representing a major release component without an additional minor release part, where “`r`” is the major release number, and with
- “`ps`” representing a minor release component modifying an major release component, where “`s`” is the minor release number.

Non-normative examples of XML Schema file names for IMS Learning Design:

- “`imsld_v1p0.xsd`” is a binding tied to IMS Learning Design v1.0,
- “`imsld_v1p0p1.xsd`” is a ‘major’ revision of the `imsld_v1p0.xsd` binding instance,
- “`imskd_v1p0p1p1.xsd`” is ‘minor’ revision of `imsld_v1p0p1.xsd` binding instance.

7.2 Namespace URI (Namespace Identifier)

The namespace URI for Learning Design shall be `http://www.imsglobal.org/xsd/imsld`.

7.2.1 Location of the Current Learning Design Binding Instance

The `.xsd` file found at `http://www.imsglobal.org/xsd/imsld/` shall be the current XML Schema binding instance of the IMS Learning Design Information Model normative elements and values. This file will change if subsequent versions of the specification are developed and released.

7.2.2 Location of Versioned Learning Design Binding Instances

Versioned instances of each binding for IMS Learning Design, including all major or minor release instances of that version’s binding, can be found at `http://www.imsglobal.org/xsd/imsld/vmpn/`, where “`vmpn`” is a folder named for a versioned instance of the specification. These instances will not change.

For example, the non-normative examples of XML Schema files named for various binding instances of IMS Learning Design v1.0 would be held at this location -- `http://www.imsglobal.org/xsd/imsld/v1p0/`.

7.3 Namespace Name

Since schemas are sometimes combined with other schemas to create documents with mixed elements, collisions between element names can sometimes occur. In order to avoid this problem, all IMS Learning Design Elements use a namespace prefix.

The namespace Name used as the prefix of an XML Namespace Qualified Name instance of an IMS Learning Design binding instance used in other XML documents shall be *imsld*.

For example, when declaring the IMS Learning Design namespace URI in an IMS Content Package manifest, this attribute representation would be used:

```
xmlns:imsld="http://www.imsglobal.org/xsd/imsld"
```

An element of the IMS Learning Design Specification declared in an XML document where the `imsld` namespace is declared would appear like this:

```
<imsld:learningDesign></imsld:learningDesign>
```

7.4 Versioning a Binding Instance

A binding version of the IMS Learning Design Specification shall always reflect the current version number of the IMS Learning Design Specification, whether or not binding structures change across specification instances. That is, the file name of the current binding instance shall share the same major and minor *version* number of the IMS Learning Design Specification.

A change to binding structures without a related change to a specification version shall result in an increment of the major or minor revision level component of a binding's file name.

Changes to binding structures without a related change in a specification shall result in an increment of the *major revision* level.

Changes to structure names or values without changes in structures themselves or the meaning associated with a value shall result in an increment of the *minor revision* level.

All versioning and revisioning shall be reflected in a binding's file name and complete version information. This new binding shall become the current binding found at <http://www.imsglobal.org/xsd/imsld>. The obsolete binding shall be moved to its place within the `imsld` file path structure based on the value of its version level component, excluding the revision level component.

7.5 Localization

7.5.1 Localization Binding Structure Name and Values

An IMS Learning Design binding's structure names and values shall be represented as depicted in the Tables within Section 3 of this document. This will ensure machine-level interoperability of learning design information.

US English annotations and comment strings within an IMS binding shall occur in all binding instances of an IMS Learning Design binding. Localized annotations and comments may be *added* to the US English annotations and comment strings within an IMS binding as deemed necessary by IMS stakeholders. Such localization activity in a binding shall promote a binding instance to the next increment in *minor* revision level.

7.5.2 Localizing this Document

This binding document may be rendered into different lingual representations except for the US English representations of

- binding structure names
- binding structure values
- file naming syntax and examples
- binding namespace URI
- binding file path representation
- binding namespace name

Localization of this document shall *not* cause any change in specification version numbering or binding revision instance numbering.

7.6 Extensibility

The IMS Learning Design Binding provides several mechanisms for extending binding instances. There is a “wildcard” element contained in <metadata> elements included in a Learning Design document. All other extensions to binding instances must be done outside of the Learning Design namespace. This means:

- The IMS Learning Design XML Schema Definition File shall not be modified;
- Elements within the IMS Learning Design namespace shall not be modified;
- All elements in an instance document that are defined to be within the IMS Learning Design namespace shall conform exactly to the definition of that element provided in this binding description document;
- Elements in another namespace shall not be substituted for elements in the IMS Learning Design namespace (via the XML Schema substitution group mechanism, for example, or via any other mechanism).

About This Document

Title	IMS Learning Design XML Binding
Editors	Rob Koper (Open University of the Netherlands), Bill Olivier (JISC CETIS), Thor Anderson (IMS), Mark Norton (IMS)
Team Co-Leads	Chuck Barritt (Cisco), Katy Campbell (University of Alberta/Industry Canada)
Version	1.0
Version Date	20 January 2003
Status	Final Specification
Summary	This document describes the XML binding of the IMS Learning Design Specification.
Purpose	Defines the IMS Learning Design XML Binding.
Document Location	http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bindv1p0.html

List of Contributors

The following individuals contributed to the development of this document:

Name	Organization
Steve Grocott	Can Studios, Ltd.
Mike Halm	Penn State University
Paul Lefrere	JISC
Jocelyn Manderveld	Open University of the Netherlands
Jon Mason	IMS Australia - DEST
Claude Ostyn	Click2Learn, Inc
Dan Rehak	Carnegie Mellon University
GT Springer	Texas Instruments
Brian Taliesin	Microsoft
Robert Todd	Digital Think
Brendon Towle	NETg
Hubert Vogten	Open University of the Netherlands
Carol Washburn	University of Pittsburgh
Bria White	Texas Instruments

Revision History

Version No.	Release Date	Comments
Base 1.0	01 April 2002	
Public Draft 1.0	18 October 2002	The first formally released version of the IMS Learning Design Specification.
Final 1.0	20 January 2003	The final version of the IMS Learning Design Schema Guide was based on the Public Draft version and updated to reflect the changes made to the schema intended to address comments raised during public draft vote and subsequent input from a variety of sources.

Index

C

Content Package 78

E

Elements

activities 12, 13, 15, 16, 20, 21,
23, 24, 38, 53, 55, 68, 74
activity-structure 13, 14, 16, 21,
23, 27, 28, 38, 42, 43,
55, 56, 63, 64, 66, 67
components 11, 13, 14, 15, 45
environments 13, 14, 16, 29, 38,
55
learning-activity 13, 16, 20, 21,
22, 28, 36, 41, 42, 43,
56, 63, 65, 66, 67, 74,
75
learning-design 11, 12, 15, 23,
30, 31, 46, 51, 66, 69,
72

notification 74, 75
roles 13, 15, 17, 18, 19, 32, 47,
69
support-activity 16, 20, 21, 26,
28, 43, 63, 65, 67, 74,
75

EML 66

Extensibility 79

I

IMS Specifications

Meta-Data 14, 52

Question and Test Interoperability 30

Reusable Definition of Competency or Education Objective 12

L

level A 25, 33, 39, 51, 53, 54, 55, 75
level B 25, 33, 39, 53, 55, 69, 74, 75
level C 55, 74
LOM 30, 47

S

Sequencing 11

U

unit of learning 14, 24, 30, 35, 37,
39, 55
URI 11, 12, 15, 17, 19, 28, 49, 50,
63, 70, 71, 72, 77, 78

V

Vocabulary 30

W

W3C 6, 8, 9, 30, 31, 47, 52, 66, 72

X

XML 6, 7, 8, 9, 10, 47, 52, 69, 70,
71, 72, 77, 78, 79
XSD 6, 8, 10

IMS Global Learning Consortium, Inc. ("IMS") is publishing the information contained in this IMS Learning Design XML Binding ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.

IMS makes no warranty or representation regarding the accuracy or completeness of the Specification.

This material is provided on an "As Is" and "As Available" basis.

The Specification is at all times subject to change and revision without notice.

It is your sole responsibility to evaluate the usefulness, accuracy, and completeness of the Specification as it relates to you.

IMS would appreciate receiving your comments and suggestions.

Please contact IMS through our website at <http://www.imsglobal.org>

Please refer to Document Name: IMS Learning Design XML Binding

Date: 20 January 2003

