1 2 3 4 5	IEEE P1484.12.3/ <u>D5</u> Draft Standard for <u>Learning Technology</u> — Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata	/ Deleted: D2
6	Sponsor	
7	Learning Technology Standards Committee	
8 9	IEEE Computer Society	
10		
11 12 13 14 15 16 17	Abstract: This <u>Standard</u> defines a World Wide Web Consortium (W3C) Extensible Markup Language (XML) Schema definition language binding of the learning object metadata (LOM) data model defined in IEEE 1484.12.1–2002 <i>Standard for Learning</i> <i>Object Metadata</i> . The purpose of this <u>Standard</u> is to allow the creation of LOM in- stances in XML. This allows for interoperability and the exchange of LOM <u>XML</u> in- stances between various systems. This <u>Standard</u> uses the W3C XML Schema definition language to define the syntax and semantics of the XML encodings.	Deleted: standard Deleted: standard Deleted: standard
18 19 20	Keywords: learning object metadata, LOM, LOM <u>XML</u> instance, LOM XML Schema binding, 1484.12.1–2002, Extensible Markup Language, XML, XML Schema definition, XSD, W3C XML Schema definition language, metadata.	
21		
22 23	The Institute of Electrical and Electronics Engineers, Inc. Three Park Avenue, New York, NY 10016-5997, USA	
24 25 26	Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Published date to be supplied. Printed in the United States of America.	Deleted: The
27 28 29 30 31 32	This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. USE AT YOUR OWN RISK! Because this is an unapproved draft, this document must not be utilized for any conformance/compliance purposes. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only. Prior to submitting this document to another standards development organization for stan-	
	Copyright © 2004 IEEE. All rights reserved.	

05 November 2004

- 33 dardization activities, permission must first be obtained from the Manager, Standards
- 34 Licensing and Contracts, IEEE Standards Activities Department. Other entities seek-
- ing permission to reproduce this document, in whole or in part, must obtain permis-35 sion from the Manager, Standards Licensing and Contracts, IEEE Standards
- 36 Activities Department. 37
- 38 **IEEE Standards Department**
- 39 Standards Licensing and Contracts
- 445 Hoes Lane, P.O. Box 1331 40 41 Piscataway, NJ 08855-1331, USA
- 42
- 43 [Note: Information about IEEE LTSC P1484.12.3 can be found at: 44
- 45 http://ltsc.ieee.org/wg12/
- 46

47 This note will be removed upon reaching the final draft of this IEEE document.]

Introduction 48

49 (This introduction is not part of P1484.12.3, Draft Standard for Learning Technology— 50 Extensible Markup Language (XML) Schema Binding for Learning Object Metadata.) Deleted: standard 51 This Standard defines World Wide Web Consortium (W3C) Extensible Markup Language Deleted: a (XML) structure and constraints on the contents of XML 1.1 documents that can be used to 52 represent learning object metadata (LOM) instances as defined in IEEE 1484.12.1-2002, 53 guage binding for 54 Standard for Learning Object Metadata. This Standard defines the structure and constraints of Deleted: data model 55 the XML 1.1 documents in W3C XML Schema definition language. 56 The purpose of this Standard is to allow the creation of interoperable LOM instances in XML. This Standard uses the W3C XML Schema definition language as the encoding. This allows 57 for interoperability and the exchange of LOM XML instances between various systems. 58

Participants 59

- 60 At the time this <u>Standard</u> was completed, the working group had the following membership:
- 61 62
 - Wayne Hodgins, Chair Erik Duval and Scott Lewis, Technical Editors Deleted: To be supplied Brandon Muramatsu Daniel Rehak Schawn Thropp Additional names to be supplied

Copyright © 2004 IEEE. All rights reserved.

Deleted: XML Schema definition lan-

Deleted: standard

ii

05 November 2004

Draft P1484.12.3/D5

63 The following persons were on the balloting committee:

Deleted: (
Deleted:)

- 64 To be provided by IEEE editor at time of publication.
- 65 Also included are the following nonvoting IEEE–SA Standards Board liaisons:
- 66 To be supplied

67 **Contents**

69	1. Overview	1
70	1.1 Scope	1
71	1.2 Purpose	1
72	2. Normative references	1
73	3. Definitions and acronyms	2
74	3.1 Definitions	2
75	3.2 Acronyms and abbreviations	4
76	4. Conformance	4
77	4.1 Strictly conforming LOM XML instances	5
78	4.2 Conforming LOM XML instances	5
79	5. LOM XML Schema binding definition	5
80	5.1 General information	5
81	5.1.1 Smallest permitted maximums	6
82	5.1.2 LOM XML instance conformance constraints	6
83	5.1.3 Extension support	6
84	5.2 LOM namespaces	7
85	5.3 Table format and organization	8
86	5.4 LOM	9
87	5.4.1 General	. 10
88	5.4.2 Life Cycle	. 12
89	5.4.3 Meta-Metadata	. 14
90	5.4.4 Technical	. 16
91	5.4.5 Educational	. 19
92	5.4.6 Rights	. 22
93	5.4.7 Relation	. 23
94	5.4.8 Annotation	. 25
95	5.4.9 Classification	. 25
96	5.5 Common data types and elements	. 27
97	5.5.1 CharacterString	. 27
98	5.5.2 DateTime	. 27
99	5.5.3 Duration	. 29
100	5.5.4 LangString	. 30
101	5.5.5 Vocabulary	. 32
102	Annex A	. 33
103	Annex B	. 34
104	Annex C	. 35
105	C.1 Composite XSDs	. 36
106	C.1.1 lomStrict.xsd	. 37
107	C.1.2 lomCustom.xsd	. 37
108	C.1.3 lomLoose.xsd	. 37
109	C.2 Component XSDs	. 38
110	Annex D	. 41

v

111	D.1 Enabling extended data elements	41
112	D.2 Enabling extended attributes	
113	Annex E	
114	E.1 Data types	
115	E.2 Elements	
116	E.3 Aggregates	
117	E.3.1 Using the XML Schema element sequence	
118	E.3.2 Using	
119	the XML Schema element all	
120	E.3.3 Using the XML Schema element choice	
121	E.3.4 Using the XML Schema element choice with the XML Schema constraint	unique
122	-	
123	E.3.5 Summary	
124	E.4 Vocabularies	50
125	E.4.1 vocab/loose.xsd	
126	E.4.2 vocab/strict.xsd	
127	E.4.3 vocab/custom.xsd	52
128	E.5 Additional notes	53
129		

130 Draft Standard for Learning Technology—

131 Extensible Markup Language (XML) Schema

- 132 **Definition Language Binding for Learning**
- 133 Object Metadata

134 **1. Overview**

135 The scope and purpose of this <u>Standard</u> are discussed in 1.1 and 1.2.

136 **1.1 Scope**

This <u>Standard</u> defines World Wide Web Consortium (W3C) Extensible Markup Language
(XML) structure and constraints on the contents of XML 1,1 documents that can be used to
represent learning object metadata (LOM) instances as defined in IEEE 1484.12.1-2002
Standard for Learning Object Metadata. ¹ This Standard defines the structure and constraint
of the XML 1.1 documents in W3C XML Schema definition language. An implementation
that conforms to this <u>Standard</u> shall conform to IEEE 1484.12.1–2002.

143 **1.2 Purpose**

The purpose of this <u>Standard</u> is to allow the creation of <u>interoperable LOM</u> instances in XML.
This <u>Standard</u> uses the W3C XML Schema definition language as the encoding. This allows
for interoperability and the exchange of LOM <u>XML</u> instances between various systems.

147 2. <u>Normative references</u>

148 The following referenced documents are indispensable for the application of this <u>Standard</u>.

For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

151 IEEE 1484.12.1–2002, Standard for Learning Object Metadata.

Copyright © 2004 IEEE. All rights reserved.

Deleted: standard

Deleted: standard
Deleted: a
Deleted: Schema definition language see
Deleted: Schema, Parts
Deleted: & 2) binding of the
Deleted: data model
Deleted: standard

Deleted:	standard

Deleted: standard

Deleted: <#>References¶

Deleted: standard

1

¹ For information on normative references, see Clause 2.

05 November 2004

152	IETF RFC 2048:1996, Multipurpose Internet Mail Extensions (MIME) Part Four: Regis-		
153	tration Procedures.		Deleted, IFTE DEC 2426-1009
154	ISO 639-1. Code for the representation of names of languages – Part 1: Alpha-2 code	1	MIME Directory Profile.¶

154 JSO 639–1, Code for the representation of names of languages – Part 1: Alpha-2 code.

- 155 ISO 639–2, Codes for the representation of names of languages Part 2: Alpha-3 code.
- ISO 3166–1, Codes for the representation of names of countries and their subdivisions Part
 1: Country codes.
- 158 ISO/IEC 10646-1, Information technology—Universal multiple-octet coded character set -
- 159 Part 1: Architecture and basic multilingual plane.
- 160 W3C Recommendation (2 May 2001), XML Schema Part 1: Structures. Deleted:
- 161 W3C Recommendation (2 May 2001), XML Schema Part 2: Data types.
- 162 <u>W3C Recommendation (4 February 2004), Namespaces in XML 1.1</u>.

3. Definitions and acronyms

164 Definitions and acronyms are defined in 3.1 and 3.2, respectively.

165 **3.1 Definitions**

For purposes of this <u>Standard</u>, the following terms and definitions apply. IEEE 100, *The Au- thoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B1]¹, should be referenced
 for terms not defined in this <u>subclause</u>.

- 169aggregate element: A LOM data element that contains other LOM data elements called170subelements. See also: subelement.
- 171 component <u>Extensible Markup Language</u> Schema definition (component XSD): An Ex-

tensible Markup Language Schema definition that <u>defines</u> a constituent of a composite
 schema. *See also:* composite Extensible Markup Language Schema definition.

- composite <u>Extensible Markup Language</u> Schema definition (composite XSD): An Exten sible Markup Language Schema definition that is a structure made up of distinct component
- 176 XML Schema definitions. *See also:* component Extensible Markup Language Schema
- 177 **definition**.
- 178 content model: A framework that identifies the makeup (i.e., data types, multiplicity con-

179 straints, ordering) of a specific model.

Copyright $\ensuremath{\mathbb{C}}$ 2004 IEEE. All rights reserved.

-1	Deleted: standard
- 1	Deleted: A1
- 1	Deleted: Clause

Deleted: XML Schema Part 1: Struc-

Deleted: XML Schema Part 2: Data

types,

Deleted:

{	Deleted: XML
-{	Deleted: represents
-{	Deleted: or part element
$\left(\right)$	Deleted: XML
-{	Deleted: XML
-{	Deleted: XML

¹ The numbers in brackets correspond to those of the bibliography in Annex A.

180	data type: A property of distinct values, indicating common features of those values and op-
181	erations on those values.

- 182 **extended data element:** A data element that is not defined in the LOMv1.0 base schema.
- 183 extended vocabulary: A value space that is not defined in the LOMv1.0 base schema for a
 LOM data element of type Vocabulary.

185 Extensible Markup Language Information Set (XML Infoset): An abstract data set that
 provides a consistent set of definitions for use in other specifications that need to refer to the
 information in a well-formed XML document (W3C, XML Information Set [B5]). See also:
 post-schema-validation infoset.

- 189 Extensible Markup Language Schema binding (XML Schema binding): A textual repre190 sentation of the behaviors, attributes, and value space of a data-model element in W3C XML
 191 Schema definition language.
- 192 Jearning object: In this <u>Standard</u>, any entity, digital or non-digital, that may be used for learn 193 ing, education, or training (IEEE 1484.12.1–2002).
- learning object metadata data element (LOM data element): A data element for which the
 name, explanation, size, ordering, value space, and data type are defined in IEEE 1484.12.1–
 2002. See also: LOMv1.0 base schema.
- learning object metadata <u>Extensible Markup Language</u> instance (LOM <u>XML</u> instance):
 A collection of metadata for a learning object that conforms to IEEE 1484.12.1–2002, that is
 represented in XML, and that adheres to the requirements and constraints of the XML <u>Schema</u>
 binding defined in this <u>Standard</u>.
- LOMv1.0 base schema: A structured collection of standard data items, including their data
 types, multiplicities, and container/component relationships, <u>defined</u> in IEEE 1484.12.1–2002,
 Clause 6.
- 204 mixed content: An element type has mixed content when elements of that type may contain
 205 character data, optionally interspersed with child elements (W3C, Extensible Markup Lan 206 guage [XML] 1.1 [B4]).
- Multipurpose Internet Mail Extensions type (MIME type): A standard way of classifying
 content types on the Internet.

209	post-schema-validation infoset: A transformed version of the XML infoset of a document
210	produced by a conformance W3C XML Schema processor. W3C XML Schema defines the
211	data elements added to the XML infoset of the original document during validation to produce
212	the post-schema-validation infoset. See also: Extensible Markup Language information
213	<u>set.</u>

1	Deleted: infoset	
1	Deleted: infoset	2
4	Deleted: [A5]	

Deleted: Extensible Markup Language
Deleted: to make the data-model element amenable to machine processing
Deleted: LangString: A data type that represents one or more character strings. A LangString value may include multiple semantically equivalent character strings, such as translations or alternative descrip- tions. <i>See also:</i> data type.¶
Deleted: standard

Deleted:	In	this	standard,	a

Deleted:	standard.	See	also:	LOMv1.0
base sche	ma			

Deleted: described

05 November 2004

- 214 subelement: A LOM data element that is contained within another LOM data element called
- 215 an aggregate element. A subelement may contain other subelements, in which case it is also an
- 216 aggregate element. See also: aggregate element.

217 token: Tokens are character strings. The value space of a token is the set of strings that do not

- 218 contain the line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces
- 219 (#x20), and that have no internal sequences of two or more spaces. The lexical space of a to-
- 220 ken is the set of strings that do not contain the line feed (#xA) nor tab (#x9) characters, that
- 221 have no leading or trailing spaces (#x20), and that have no internal sequences of two or more
- 222 spaces. The base type of a token is normalizedString. For purposes of this Standard, to-223 kens are case sensitive. (Adapted from W3C, XML Schema Part 2.) See also: token set.

Deleted: normalizedString (adapted from XML Schema Part 2: Data types) Deleted: standard

- 224 token set: A set of tokens in which each token is unique. See also: token.
- 225 uniqueness constraint: A restriction placed on a LOM data element that enforces that the
- 226 LOM data element is unique within a LOM XML instance. If a LOM data element has a uniqueness constraint, it appears in a LOM XML instance zero or one time. See also: learning 227
- object metadata data element. 228
- 229 value space: The set of values for a given data type (ISO/IEC 11404:1996 [B3]).
- NOTE-In IEEE 1484.12.1-2002, a value space is typically enumerated outright or defined by 230
- 231 reference to another standard or specification.

3.2 Acronyms and abbreviations 232

- 233 IANA: Internet Assigned Numbers Authority
- 234 LOM: Learning Object Metadata
- 235 MIME: Multipurpose Internet Mail Extensions
- 236 PSVI: post-schema-validation infoset
- 237 SPM: smallest permitted maximum
- 238 UTC: coordinated universal time
- 239 W3C: World Wide Web Consortium
- 240 XML: Extensible Markup Language
- 241 XSD: XML Schema definition

4. Conformance 242

243	<u>Conformance to this Standard is discussed in 4.1 and 4.2. In 4.1 and 4.2. strictly conforming</u>
244	<u>LOM XML instance and conforming LOM XML instance refer to the metadata represented in</u>
245	the LOM XML instance prior to any processing of the LOM XML instance.
246 247	In this Standard, "shall" is to be interpreted as a requirement on an implementation; "shall not" is to be interpreted as a prohibition.

Copyright © 2004 IEEE. All rights reserved.

Deleted: [A4]

Deleted: This standard provides definitions of W3C XML Schema definition language structures used to produce both strictly conforming and conforming LOM instances.¶

In this standard, "shall" is to be interpreted as a requirement on an implementation; "shall not" is to be interpreted as a prohibition.¶

<#>Strictly conforming LOM instances¶

A strictly conforming LOM instance ¶ <#>Shall conform to the LOMv1.0 base schema requirements of IEEE 1484.12.1-2002:¶

Shall conform to the requirements of Clause

Deleted: ;

Deleted: ¶

4

<#>Shall not contain any extensions to the LOMv1.0 base schema defined in IEEE 1484.12.1-2002. <#>Conforming LOM instances¶ A conforming LOM instance ¶ <#>Shall conform to the LOMv1.0 base schema requirements of IEEE 1484.12.1-2002; and ¶ Shall conform to the requirements of Clause

I

248	4.1	Strictly conforming LOM XML instances
249	<u>A stri</u>	ctly conforming LOM XML instance
250	_	Shall be a strictly conforming LOM instance as defined in IEEE 1484.12.1-2002;
251	_	Shall conform to the requirements of Clause 5 of this Standard;
252 253	_	Shall not include vocabulary values that are not defined in Clause 5 of this Stan- dard;
254 255	_	Shall not include XML elements or attributes that are not defined in Clause 5 of this Standard; and
256	_	Shall not include mixed content.
257	4.2	Conforming LOM XML instances
258	<u>A con</u>	forming LOM XML instance
259	_	
		Shall be a conforming LOM instance as defined in IEEE 1484.12.1–2002;
260	_	Shall be a conforming LOM instance as defined in IEEE 1484.12.1–2002; Shall conform to the requirements of Clause 5 of this Standard;
260 261	_	<u>Shall be a conforming LOM instance as defined in IEEE 1484.12.1–2002;</u> <u>Shall conform to the requirements of Clause 5 of this Standard;</u> <u>May include vocabulary values that are not defined in Clause 5 of this Standard;</u>
260 261 262 263		Shall be a conforming LOM instance as defined in IEEE 1484.12.1–2002; Shall conform to the requirements of Clause 5 of this Standard; May include vocabulary values that are not defined in Clause 5 of this Standard; May include XML elements and attributes that are not defined in Clause 5 of this Standard; Standard by using the extension mechanism described in 5.1.3; and

265 **5. LOM XML Schema binding definition**

The LOM XML Schema binding is defined in 5.1 – 5.5. Subclause 5.4 lists the LOM data
elements defined in the LOMv1.0 base schema. For each LOM data element, the corresponding XML element for the XML Schema binding is described.

269 **5.1 General information**

270	As defined in IEEE 1484.12.1-2002, all LOM data elements are optional. An XML Schema
271	definition (XSD) for the LOMv1.0 base schema shall not require any LOM data to be present
272	in a LOM <u>XML</u> instance.

273The LOMv1.0 base schema defines an aggregation relationship between data items. An XSD274for LOM defines an aggregation relationship between subelements that maintains the relation-275ship defined in the LOMv1.0 base schema. In a LOM XML instance, a subelement shall ap-

276 pear only within the <u>aggregate</u> element of the aggregation relationship. For example, in

277 5.4.3.1, the Identifier subelement appears by definition as a component of the Meta-Metadata

Copyright © 2004 IEEE. All rights reserved.

1	Deleted: of
-{	Deleted: W3C
$\left(\right)$	Deleted: definition language element for XML

Deleted: parent

5

element described in 5.4.3. The presence of the subelement automatically implies the presence of the aggregate element to which the <u>subelement</u> belongs.

280 The LOMv1.0 base schema does not define any <u>sequencing</u> of LOM data elements, except for

281 their aggregation relationships. Therefore, an XML Schema binding of the LOMv1.0 base

282 schema shall not define any sequence of the elements in a LOM <u>XML</u> instance, except for

their aggregation relationships.

NOTE—XML names used in this Standard are derived from LOM data element names by con catenating the LOM data element names and using lowerCamelCase capitalization.

286 **5.1.1 Smallest permitted maximums**

The W3C XML Schema definition language does not support the concept of smallest permitted maximums (SPMs) as defined in IEEE 1484.12.1–2002. The W3C XML Schema definition language does provide a means for restricting maximum lengths (i.e., maxLength for character strings) and maximum numbers of occurrences (i.e., maxOccurs for <u>aggregate</u> elements); however, these restrictions are not compatible with the definition of SPM.

If a LOM XML instance contains more than the SPM number of occurrences of a LOM data
 element, then an application shall process at least the SPM number of occurrences of the LOM
 data element. If a LOM XML instance contains more than the SPM number of characters in a
 character string, then an application shall process at least the SPM number of characters in the
 character string. (SPMs for LOM data elements and character strings are defined in IEEE
 1484.12.1–2002.)

NOTE—To encourage interoperability, creators of XSDs intended to validate LOM XML in stances against IEEE 1484.12.1–2002 should not use maxOccurs or maxLength restrictions
 or should set the restricted limit to unbounded.

301 5.1.2 LOM XML instance conformance constraints

Constraints on the validity of a *conforming* or *strictly conforming* LOM XML in stance with respect to IEEE 1484.12.1–2002 are defined throughout 5.4 and 5.5. All *conform- ing* and *strictly conforming* LOM XML instances shall satisfy these constraints.

NOTE—Some applications may use an XSD to validate a LOM XML instance. If an application uses an XSD to validate a LOM XML instance, the post-schema-validation infoset (PSVI) of a LOM XML instance is insufficient to determine the validity of the LOM XML instance with respect to IEEE 1484.12.1–2002. Use of an XSD to validate a LOM XML instance does not relieve an application from any requirements on enforcing the constraints that appear throughout 5.4 and 5.5.

311 **5.1.3 Extension support**

312 Extensions to the LOMv1.0 base schema are permitted by this Standard. As defined in IEEE
 313 1484.12.1–2002, extensions to the LOMv1.0 base schema shall retain the data types and value

Copyright © 2004 IEEE. All rights reserved.

Deleted:	component

Deleted: sequence

Deleted: W3C	
Deleted: Schema definition language element	
Deleted: are encoded in	_

Deleted: not provide a mechanism to specify the minimum number of elements in composite data types. The W3C XML Schema definition language does

Deleted: composite elements). However

Deleted: NOTE—To encourage interoperability, creators of XSDs intended to validate LOM instances against IEEE 1484.12.1–2002 should not use maxOccurs or maxLength restrictions.¶ <#>Post-parsing validation¶ The post-schema validation XML infoset

Deleted: instance is insufficient to determine its validity with respect to IEEE 1484.12.1–2002. After a LOM instance is validated against the XSD, it

Deleted: be subjected to additional postparsing validation rules. These additional rules

Deleted: A strictly conforming XML binding of the LOMv1.0 base schema shall not support extensions to the LOM data elements.¶ A conforming XML binding of the LOMv1.0 base schema may include extensions to the LOM data elements. Extensions shall not conflict with the W3C XML Schema definition language names and namespaces defined in this standard.¶

314 315 316 317 318 319 320 321 322 323	 spaces of LOM data elements from the LOMv1.0 base schema and shall not define data types or value spaces for aggregate elements in the LOMv1.0 base schema. This Standard permits the following types of extensions for <i>conforming</i> but not for <i>strictly conforming</i> XML bindings: LOMv1.0 base schema XML element extension: LOMv1.0 base schema data elements may be extended. These extensions are new XML elements defined in a namespace other than the namespaces defined in 5.2. Extended data elements may be added to any LOM data element that is defined as an aggregate element. Extended data elements shall not be added to LOM data elements that are not aggregate elements (see 5.4 and 1.1). 		
324 325 326 327	 <u>LOMv1.0 base schema XML attribute extension: LOM data elements may be extended by defining attributes for the LOM data elements. These extensions are XML attributes defined in a namespace other than the namespaces defined in 5.2. These attributes may be added to any LOM data element (see 5.4 and 5.5).</u> 		Deleted: The following namespaces are reserved by this standard
328 329 330 331 332	 LOMv1.0 base schema vocabulary data type extension: LOMv1.0 base schema vocabularies may be extended. These extensions are additional tokens for LOM data elements of type Vocabulary (see 5.4). If vocabularies are extended, the source of the additional vocabulary tokens should be identified and shall not be LOMv1.0. 		Deleted: used in defining the elements Deleted: this standard:¶ The Deleted: name for this standard shall be: Deleted: standard,
333 334	5.2 LOM namespaces <u>All of the LOM data elements defined in 5.4 and 5.5 shall be defined in the namespace</u>		Deleted: :¶ Deleted: The namespace name prefix for the http://ltsc.ieee.org/xsd/LOM namespace shall be lom. This value is
335 336 337 338	shall not be used to represent any other data element (see W3C Recommendation, Name- spaces in XML 1.1.). This namespace, shall not be used to define extensions to a LOM XML instance.		reserved by this standard, Deleted: the prefix Deleted: namespace; Deleted: The namespace
339 340 341	The following namespaces are reserved by this Standard for future use and shall not be used to represent any other data element. The namespaces are used in the XSDs provided with this Standard (see Annex B).		Deleted: The namespace Deleted: shall be reserved for the LOM XML custom composite XSDs (see An- nex B) provided by IEEE;
342	<pre>- http://ltsc.ieee.org/xsd/LOM/custom,</pre>		Deleted: fine nancespace Deleted: shall be reserved for the LOM XML unique composite XSDs (see An- nex B) provided by IEEE;
343	- http://ltsc.ieee.org/xsd/LOM/unique	1	Deleted: The namespace
344 345	 http://ltsc.ieee.org/xsd/LOM/vocab	/	Deleted: shall be reserved for the LOM XML vocab composite XSDs (see Annex B), provided by IEEE; and
346	If an organization provides extensions to a LOM XML instance, the organization should de-		Deleted: The namespace
347	fine the namespace for these extensions.		Deleted: shall be reserved for the LOM XML extension composite XSDs (see Annex B) provided by IEEE.

Deleted: NOTE— Deleted: extension elements

Deleted: the Deleted: is free to

348 **5.3** <u>Table</u> format and organization

This <u>Standard</u> uses <u>tables</u> to describe the requirements for the LOM <u>XML</u> instances. The ta bles express requirements for each of the LOM data elements in the LOMv1.0 base schema. These requirements are

LOM <u>data element</u>: The name of the LOM data element in the LOMv1.0 base schema.

- *XML name:* The <u>name used in a LOM XML instance that conforms to this Standard for the corresponding LOM data element. A dash ("-") in the tables in 1.1, which describe common LOM data types and LOM data elements, indicates that no XML name exists because the LOM name refers to a data type instead of a LOM data element.
 </u>
- 359 Subelements: A listing of subelements of the LOM data element. An entry of
 "None" indicates that the LOM data element does not have subelements. If present, the subelements listed shall be contained by their associated aggregate element. This ensures that the aggregation relationships of the LOMv1.0 base
 schema components are enforced. The order of appearance of subelements shall
 not be significant.
- Min: If a LOM data element is not the top-level LOM element, the requirements on the minimum number of times the LOM data element may appear in a LOM XML instance in the context of the LOM data element's aggregate element. For the LOM element, the requirements on the minimum number of times the LOM data element may appear in a LOM XML instance.
- *Max:* <u>If a LOM data element is not the top-level LOM element, the requirements</u>
 on the maximum number of times the <u>LOM data element may appear in a LOM</u>
 XML instance, in the context of the <u>LOM data element's aggregate element</u>. For
 the LOM element, the requirements on the maximum number of times the LOM
 element may appear in a LOM XML instance. An <u>"∞"</u> indicates that the maximum number of times is unbounded. Values in parenthesis are SPMs.
- Order: Indicates whether the order of the values is significant. This <u>Standard uses</u>
 three designators for the order: "Ordered", "Unordered", and "Unspecified", "<u>Or</u>
 dered" indicates that the ordering of the values is significant. "Unordered" indicates that the ordering of the values is not significant. For a LOM data element
 that has a multiplicity of zero or one, the concept of order has no meaning, which
 is indicated by the designator "Unspecified". Note: The significance of the order
 of a list of ordered values is determined by the implementation.
- *LOM data type;* Indicates whether the LOM data type is LangString, DateTime, Duration, Vocabulary, or CharacterString, or is Unspecified. If the LOM data element is an aggregate element, it has no data type and, therefore, is described as "Unspecified". If the data type is Vocabulary, the permissible values from the LOMv1.0 base schema are listed.

Deleted: Tabular
Deleted: table
Deleted: standard
Deleted: a tabular format

Deleted: name	
Deleted: element	
Deleted: hinding of	

Deleted: The
Deleted: sub
Deleted: ,
Deleted: sub
Deleted: The
Deleted: sub
Deleted: ,

Deleted:	standard

Deleted: sub

Deleted: "n"

Deleted: ¶
Data type: Indicates whether the

Deleted: are LangString, DateTime, Duration, Vocabulary, CharacterString, or Unspecified (Unspecified indicates that the data element

Deleted:).

8

388 Each table that describes an aggregate element includes the direct-descendant subelements of 389 that aggregate element. For example, Table 1 in 5.4 describes the LOM data element and in-390 cludes the LOM data element's direct-descendant subelements, such as General and Life Cy-391 cle. However, the table does not include the subelements of the General element, such as 392 Identifier and Title. If a subelement is also an aggregate element, a separate table describes the 393 subelement with its direct-descendant subelements. If information beyond that provided by 394 IEEE 1484.12.1-2002 is needed to define subelements in a LOM XML instance, that informa-395 tion is provided in subclauses of the aggregate element description. 396 With the exception of the order for Description, Educational, Identifier, and Intended End User Role, in the case of any discrepancy in the tables in 5.4 and 5.5 and IEEE 1484.12.1-397

398 2002, the values from IEEE 1484.12.1–2002 shall be used.

399Although the Max column may indicate that a subelement may appear in an aggregate ele-
ment an unbounded maximum number of times, an application that processes a LOM XML

401 <u>instance shall process at least the SPM number of subelements (see 5.1.1).</u>

402 **5.4 LOM**

403

Table 1 describes the LOM element and its <u>direct-descendant</u> subelements.

404

Table 1—The LOM element

LOM <u>data</u>	XML-name	Subelements	- Min-	Max -	Order	- <u>LOM data</u> type-
LOM	lom	_General	1	_1	Unspecified	_Unspecified
		Life Cycle				
		Meta-Metadata Technical				
		Educational				
		Rights				
		Relation				
		Annotation				
		Classification				
General	general	<u>See 5.4.1</u>	<u>0</u>	_1	Unspecified	_Unspecified
Life Cycle	lifeCycle	<u>See 5.4.2</u>	<u> </u>	_1	Unspecified	_Unspecified
Meta-Metadata	metaMetadata	<u>See 5.4.3</u>	<u>0</u>	_1	Unspecified	Unspecified
Technical	technical	<u>See 5.4.4</u>	<u>0</u>	_1	Unspecified	Unspecified
Educational	educational	<u>See 5.4.5</u>	<u>0</u>	_	Unordered	Unspecified
				<u>(100)</u>		
Rights	rights	<u>See 5.4.6</u>	<u>0</u>	1	Unspecified	Unspecified
Relation	relation	<u>See 5.4.7</u>	<u>0</u>	-	Unordered	Unspecified
				<u>(100)</u>		



	LOM data	X ML-name	Subelements	- Min-	- Max -	Order	-LOM data type-		Deleted: name		
	<u>element</u>		Subtienents		17102	oruer	Point and the		Deleted: Data		
	Annotation	annotation	<u>See 5.4.8</u>	_0	- <u>*</u>	Unordered	Unspecified		Deleted: <		
	Glassification	-1:6:+:	Sec 5.4.0	0	(30)	Theredourd	Linen esifie d		Deleted: >		
	Classification	classification.	<u>See 5.4.9</u>	= ≌ = =	$=\frac{\infty}{(40)}$	Unordered		γ	Deleted: Entity¶		
									Description		
405	Namespace declaration										
406	The LOM XML instance shall include a namespace declaration that declares the LOM name-										
407	space for the	ne LOM elemen	t and its componen	its. T	he LC	DM names	bace shall be		Deleted: >		
408	"http://lt		sd/LOM" as defined in	n 5.2			-		Deleted: Purpose¶		
				•					Description		
409	The namespa	ce declaration use	d in <u>conformance with</u>	<u>"W3</u>	C Reco	ommendatio	n <u>, Namespaces</u>		Keyword		
410	in XML <u>1.1."</u>	Examples are show	wn in Figures 1 and 2,	respec	tively				Deleted: n		
411	Deleted:										
411	Deleted: an XML instance can be de-										
412 413	clom xmlng="http://ltgg_iooo_org/xgd/IOM">										
414	· · ·	- 1100207/10000	1000.019/xbu/10M					$\frac{1}{\lambda} = \frac{\lambda}{\lambda}$	Deleted: instance, which may be an		
415									element other than the LOM element, or		
410	by declaring the LOMv1.0 base schema namespace within the LOM element each										
417	Figure 1—An example default namespace declaration										
								1	The namespace declaration shall take the		
418	form of a default namespace declaration or a prefix-specific namespace declara-										
419	<lom:lom td="" xr<=""><td>tion in conformance with W3C XML Recommendations</td></lom:lom>	tion in conformance with W3C XML Recommendations									
420	<pre> </pre>			Deleted: :							
422											
423	Fie	oure 2—An exa	nple prefix-specific	nam	espac	e declarat	tion				
		J									
40.4									Deleted: is required so that applications		
424	NOTE—A namespace declaration allows an application to recognize a LOM XML instance as will will										
423	one that conta	ins LOW data elem	ents described in this <u>St</u>	andaro	<u>1</u>			-	Deleted: LOM data as		
101	E A A Com	aral							Deleted: that conforms to		
426	5.4.1 Gen	eral							Deleted: standard		
427	Table 2 descr	ibes the General el	ement and its direct-de	scend	ant sub	elements		11	Deleted: first generation		
,	1 4010 2 40501			200110				2			

428

Table 2—The General element

LOM, <mark>data</mark>	XML name	– – Subelements – –		- Max-	Order	I-OM data type-		Deleted: name
<u>element</u>		Subtrements			oruer			Deleted: Data*
General	general	Identifier	0	1	Unspecified	Unspecified		Deleted: <>
		Title						
		Language						
		Description						
		Keyword						
		Coverage						
		Structure						
		Aggregation Level						
Identifier	identifier	See 5.4.1.1	0	ø	Unordered	Unspecified		Deleted: <>
				<u>(10)</u>				Deleted: Unspecified
Title	title	<u>See 5.5.4</u>	0	1	Unspecified	LangString		Deleted: n
Language	language	None	0	ø	Unordered	CharacterString		Deleted: <>
				<u>(10)</u>				Deleted: <>
Description	description	See 5.5.4	0	<u>x</u>	Unordered	LangString	ì	Deleted: n
				(10)			- ```	Deleted: <>
Keyword	keyword	<u>See 5.5.4</u>	0	(10)	Unordered	LangString	Ì	Deleted: n
Coverage	coverage	See 5.5.4	0	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Unordered	I anoStrino		Deleted: <>
coverage		000 0.0.1		(10)	Choracted			Deleted: n
Structure	structure	See 5.5.5	0	1	Unspecified	Vocabulary		Deleted: <>
							Ň	Deleted: n
Aggregation	aggregationLevel	See 5.5.5	0	1	Unspecified	Vocabulary		Deleted: <>
Level								Deleted: <>
		I	1	1	I		/	Deleted: *Common data types and data elements are defined in . ¶
5.4.1.1 100							,	Deleted: first-generation

429 430

Table 3 describes the Identifier element and its <u>direct-descendant</u> subelements.

431

Table 3—The Identifier element

LOM <mark>data</mark>	XML name	Subelements	– – Min – –	- Max-	Arder	I-OM data-type		Deleted: name
<u>element</u>		Bubelements	10111	What	oruci	Point data type		Deleted: Data*
Identifier	identifier	Catalog	0	₽	Unordered	Unspecified		Deleted: <> [16]
		Entry		<u>(10)</u>				Deleted: Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString	Ì,	Deleted: n
Entry	entry	None	0	1	Unspecified	CharacterString		Deleted: <>> [17]
	•							Deleted: <>

432 5.4.1.2 Language

433 The Language element shall be a character string where the value of the character string shall 434 be one the following:

435	- The format and values described in 5.5.4.1; or	Deleted: the
-55		Deleted: 5.51
436	– <u>The</u> token none.	Deleted: the

437 5.4.1.3 Structure

438 If the value space for the Source subelement of the Structure element is the LOMv1.0 base 439 schema (i.e., <source>LOMv1.0</source>), then the valid values for the Value subele-440 ment of the Structure element shall come from the following list of tokens:

- 441 atomic
- 442 collection _
- 443 networked
- 444 hierarchical
- 445 linear

5.4.1.4 Aggregation Level 446

447 If the value space for the Source subelement of the Aggregation Level element is the 448 LOMv1.0 base schema (i.e., <source>LOMv1.0</source>), then the valid values for the 449 Value subelement element of the Aggregation Level elements shall come from the following list of tokens: 450

- 451 1
- 452 2
- 453 3
- 454 4

5.4.2 Life Cycle 455

Table 4 describes the Life Cycle element and its direct-descendant subelements. 456

457

Table 4—The Life Cycle element

l	LOM data	X ML name	Subelements	Min	- Max -	Order	I-OM data-type		Deleted: name
l	<u>element</u>		Subtrements	1VIII	Max	oruer	Port data ope	57	Deleted: Data
ļ	Life Cycle	lifeCycle	Version	_0	_1	Unspecified	_Unspecified		Deleted: *
			Status						Deleted: <lifecycle></lifecycle>
			Contribute						

Copyright © 2004 IEEE. All rights reserved.

Deleted: source of the Deleted: <\ Deleted: value

Deleted: first-generation

Deleted: shall be one

12

Deleted: *Common data types and data	ι
elements are defined in . ¶	

Deleted: the	
Deleted: 5.51	
Deleted: the	

Deleted: source of the
Deleted: <\
Deleted: value
Deleted: be one of

LOM, <u>data</u> <u>element</u>	XML name	Subelements	Min	Max -	Order	<mark>J-OM data</mark> type
Version	version	<u>See 5.4.4</u>	_0	1	Unspecified	LangString
Status	status	<u>See 5.5.5</u>	_0	_1	Unspecified	_ <u>Voca</u> bulary
Contribute	contribute	<u>See 5.4.2.2</u>	_0	<u></u>	Ordered	_Unspecified
				<u>(30)</u>		

458 **5.4.2.1 Status**

459 If the value space for <u>Source subelement of the Status element is the LOMv1.0</u> base schema
460 (i.e., <source>LOMv1.0
461 <u>Status element shall come from the following list of tokens:</u>

- 462 draft
- 463 final
- 464 revised
- 465 unavailable

466 **5.4.2.2 Contribute**

467 Table 5 describes the Contribute element and its <u>direct-descendant</u> subelements.

468

Table 5—The Contribute element

LOM <u>,data</u> <u>element</u>	XML name	Subelements	-Min-	- Max -	Order	<u>LOM data type</u>
Contribute	contribute	<u>Role</u>	_0	<u>∞</u>	Ordered	
		Date				
Role	role	<u>See 5.5.5</u>	_0	1	Unspecified	Vocabulary
Entity	entity	None	_0	- <u>*</u>		<u>CharacterString</u>
Date	date	<u>See 5.5.2</u>	_0	_1	Unspecified	DateTime

469 **5.4.2.2.1 Role**

470 If the value space for the <u>source subelement of the Role</u> element is the LOMv1.0 base schema
471 (i.e., <source>LOMv1.0
475 (i.e., <source>LOMv1.0
476 (i.e., <source>LOMv1.0
476 (i.e., <source>LOMv1.0
476 (i.e., <source>LOMv1.0
477 (i.e., <source>LOMv1.0
478 (i.e., <source>LOM

472 <u>Role</u> element shall <u>come from</u> the following <u>list of</u> tokens:

473 - author

474 - publisher

475 – unknown

1	Deleted: name
+	Deleted: Data
1	Deleted: *
1	Deleted: <
1	Deleted: >
Ì	Deleted: String
Ì	Deleted: <
Ì	Deleted: >
Ì	Deleted: Source¶ Value
Ì	Deleted: <
ij	Deleted: >
	Deleted: Role¶ Entity¶ Date
ļ	Deleted: n
j	Deleted: *Common data types a [19]
j	Deleted: source of the
j	Deleted: <\
ļ	Deleted: value
1	Deleted: be one of
+	Deleted: first-generation
,1	Deleted: name
.1	Deleted: Data
1	Deleted: *
1	Deleted: <
+	Deleted: >
1	Deleted: n
1	Deleted: <
-	Deleted: >
1	Deleted: Source¶ [20]
1	Deleted: <
1	Deleted: >
Ì	Deleted: vCard
ľ	Deleted: n
	Deleted: <
ſ,	Deleted: >
	Deleted: DateTime¶ [21]
,1	Deleted: *Common data types a [22]
,1	Deleted: source of the
	Deleted: <\
	Deleted: value
1	Deleted: be one of

476	- initiator
477	- terminator
478	- validator
479	- editor
480	- graphical designer
481	- technical implementer
482	- content provider
483	- technical validator
484	- educational validator
485	- script writer
486	- instructional designer
487	- subject matter expert
488	5.4.2.2.2 <u>Entity</u>
489 490	The value held by the Entity element shall be a character string literal that is the canonical lexical representation of a valid vCard as defined in IETF RFC 2426:1998.
491 492	NOTE—IETF RFC 2426:1998 does not rely on XML syntax to express the internal structure of a valid vCard. At the time of publication of this Standard, no <i>de facto</i> standard W3C XML Schema

493 definition language binding for the vCard specification existed.

494 **5.4.3 Meta-Metadata**

- 495 Table 6 describes the Meta-Metadata element and its <u>direct-descendant</u> subelements.
- 496

Table 6—The Meta-Metadata element

LOM <u>data</u>	XML name	Subelements	Min	- Max -	Order	<mark>LOM data</mark> type	
Meta-Metadata	metaMetadata	Identifier	_0	1	Unspecified	Unspecified	ľ.
		Contribute					
		Metadata					4
		Schema					$-\frac{y_j'}{y_j'}$
		Language					
Identifier	identifier	See 5.4.3.1	_0	<u></u>	Unordered	_Unspecified	ľ,
				<u>(10)</u>			11
Contribute	contribute	<u>See 5.4.3.2</u>	_0	<u></u>	Ordered	Unspecified	ľ
				<u>(10)</u>			
Metadata Schema	metadataSchema	None	0	<u></u>	Unordered	CharacterString	<
				<u>(10)</u>			[```

Deleted: first-generation
Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: <
Deleted: >
Deleted: Catalog¶ Entry
Deleted: Unspecified
Deleted: n
Deleted: <
Deleted: >
Deleted: Role¶ Entity¶ Date
Deleted: n
Deleted: <
Deleted: >
Deleted: n

element	XMŁ name-	Subelements-	Min	- Max -	Order	<u>LOM data</u> ty
Language	language	None	_0	_1	Unspecified_	CharacterString
5.4.3.1 Jde Table 7 descr	ntifier ribes the Identific	er element and its, Fable 7—The Id	direct-des	<u>cendant</u> elemen	subelements t	3
LOM <u>data</u> <u>element</u>	XML name		Min	- Max-	Order	J.OM data ty
Identifier	identifier	<u>Catalog</u> Entry		$-\underbrace{+}{+} \underbrace{+}{+} \underbrace{+}{(\underline{10})} = = =$		Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString
	entry	None	0	1	Unspecified_	CharacterString
5.4.3.2 Cor Table 8 descr	ribes the Contrib	ute element and it	s <u>direct-de</u>	elemei	<u>ut</u> subelemen nt	ts.
Entry 5.4.3.2 Cor Table 8 descr LOM data element	ntribute ribes the Contrib Ta	ute element and it able 8—The Co	s <u>direct-de</u> ontribute	elemei - Max -	nt Order -	ts. J.OM data ty
Entry 5.4.3.2 Cor Table 8 descr LOM data element	ribes the Contrib Ta	ute element and it able 8—The Co Subelements- Role Entity	s <u>direct-de</u>	elemei - Max - $\frac{\varphi_{2}}{(10)} = $	nt Order -	ts. <u>LOM data</u> ty
Entry 5.4.3.2 Çor Table 8 descr LOM data element Contribute	ribes the Contrib Ta	ute element and it able 8—The Co Subelements- Role Entity Date	s <u>direct-de</u> ontribute	elemei Max - - $\frac{42}{(10)} = =$	nt Order	ts
Entry 5.4.3.2 Cor Table 8 descr LOM data element Contribute	role	ute element and it able 8—The Co Subelements- Role Entity Date See 5.5.4	s <u>direct-de</u> ontribute Min	elemei $ Max + \frac{2}{(10)} = =$	t subelemen nt Ordered Unspecified	ts. <u>LOM data</u> ty <u>Unspecified</u>
Entry 5.4.3.2 Cor Table 8 descr LOM data element Contribute Role Entity	ribes the Contrib Ta	ute element and it able 8—The Cc Subelements Role Entity Date See 5.5.4 None	s <u>direct-de</u> ontribute Min 0 0 0	elemei - Max - $\frac{\varphi_{2}^{2}}{(10)} = =$	nt Order - Ordered	ts
Entry 5.4.3.2 Cor Table 8 descr LOM data element Contribute Role Entity Date	ribes the Contrib Ta Contribute contribute role entity date	ute element and it able 8—The Co Subelements- Role Entity Date See 5.5.4 None	s <u>direct-de</u> ontribute Min 0 0 0 0 0	elemei Max - $- \frac{2}{10} = =$ $\frac{1}{2} = =$	t subelemen nt Ordered Unspecified Unspecified	ts. <u>LOM data</u> ty <u>Unspecified</u> = = <u>Vocabulary</u> = = <u>CharacterString</u> <u>DateTime</u> = =
Entry 5.4.3.2 Cor Table 8 descr LOM data element Contribute Role Entity Date 5.4.3.2.1 Ro	ribes the Contrib Ta 	ute element and it able 8—The Cc Subelements Role Entity Date See 5.5.4 None See 5.5.2	s <u>direct-d</u> ontribute Min - 0 - 0 - 0 - 0	elemei elemei Max - $- \frac{42}{(10)} = =$ $- \frac{42}{(10)} = =$ $- \frac{42}{(10)} = =$	t subelemen nt Ordered Unspecified	ts. - <u>LOM data</u> ty = Unspecified = = = <u>Vocabulary</u> = = = <u>CharacterString</u> = <u>DateTime</u> = =

Deleted: Data Deleted: * Deleted: < Deleted: > Deleted: *Common data types a ... [23] Deleted: first-generation Deleted: name **Deleted:** Data Deleted: * Deleted: < Deleted: > Deleted: Unspecified Deleted: n Deleted: < Deleted: >> Deleted: < Deleted: > Deleted: vCard Deleted: *Common data types a ... [24] Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: < Deleted: > Deleted: n Deleted: < Deleted: > Deleted: Source ... [25] Deleted: < Deleted: > Deleted: vCard Deleted: N Deleted: < Deleted: > Deleted: DateTime¶ ... [26] Deleted: *Common data types a [27] Deleted: source of the Deleted: <\ Deleted: value Deleted: be one of

Deleted: name

509 5.4.3.2.2 Entity

- The value held by the Entity element shall be a character string literal that is the canonical 510 lexical representation of a valid vCard as defined in IETF RFC 2426:1998. 511
- 512 NOTE-IETF RFC 2426:1998 does not rely on XML syntax to express the internal structure of a
- 513 valid vCard. At the time of publication of this Standard, no de facto standard W3C XML Schema
- 514 definition language binding for the vCard specification existed.

515 5.4.3.3 Metadata Schema

- 516 For those LOM XML instances that are *strictly conforming* to this Standard (see 4), if the 517 Metadata Schema element (metadataSchema) is present in a LOM XML instance, the ele-518 shall value (i.e., ment contain the LOMv1.0 519 <metadataSchema>LOMv1.0</metadataSchema>)
- 520 For those LOM XML instances that are *conforming* but not *strictly conforming* (i.e., contain
- 521 extended data model elements, see 4), if the Metadata Schema element (metadataSchema)
- 522 is present in a LOM XML instance, the element shall contain the value LOMv1.0, and addi-
- 523 tional occurrences of metadataSchema should list all other metadata schemas or authorita-
- 524 tive specifications used to create the LOM XML instance.

525 5.4.3.4 Language

526 The Language element shall be a character string where the value of the character string shall have the format and value space described in 5.5.4.1. 527

5.4.4 Technical 528

- Table 9 describes the Technical element and its direct-descendant subelements. 529
- 530

Table 9—The Technical element

LOM, <u>data</u> <u>element</u>	XML-name	Subelements -	-Min-	- Max-	Order	-LOM data type,-	14
Technical	technical.	<u>Format</u>	_0	_1	Unspecified	Unspecified	
		Size					
		Location					
		Requirement					
		Installation Remarks					
		Other Platform Requirements					
		Duration					

Deleteur · Shan	
Deleted: be	
Deleted: values	
Deleted: first-generation	

Deleted: name **Deleted:** Data Deleted: * Deleted: < Deleted: >

Convright © 2004 IEEE	All rights reserved
copyright @ 2004 ILLL.	I in fights reserved.

Deleted: standard
Deleted: 4
Deleted: <
Deleted: >
Deleted: .¶
Deleted: Instances
Deleted: 4
Deleted: <
Deleted: >
Deleted: .
Deleted: > shall

Deleted: name Deleted: Data Deleted: * Deleted: * Deleted: n Deleted: * Deleted: * Deleted: * Deleted: n Deleted: n Deleted: *

Deleted: OrComposite

Deleted: n

Deleted: <

Deleted: >
Deleted: <

Deleted: >

Deleted: <

Deleted: >

tion of

literal
Deleted: the

Deleted: s

Deleted: *Common data types and data

Deleted: canonical lexical representa-

Deleted: shall be a character string

elements are defined in . ¶

Deleted: first-generation
Deleted: Data

Deleted: OrComposte

Maximum Version
Deleted: n

Deleted: first-generation

Deleted: name Deleted: <

Deleted: >

Deleted: n Deleted: < Deleted: > Deleted: Type¶ Name¶ Minimum Version¶

LOM <u>data</u> <u>element</u>	XML name		-Min-	- Max-	Order	LOM data type
Format	format	None	_0	<u>≠∞</u> (40)	Unordered	CharacterString
Size	size	None	_0	_1	Unspecified	CharacterString
Location	location	None	_0	<u>≠∞</u> (10)	Ordered	CharacterString
Requirement	requirement.	<u>See 5.4.4.3</u>	_0	 ≠ = = = = (40)	Unordered	Unspecified
Installation Remarks	installationRemarks	<u>None</u>	_0	1	Unspecified	LangString
Other Platform Requirements	otherPlatformRequirements	None	_0	_1	Unspecified	LangString
Duration	duration	None	_0	1	Unspecified	Duration
The <u>value h</u> acter string - <u>A 1</u> Mai	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v	hall be a chara lexical repres alue, from RFC	<u>sentati</u> 2 2048	on of	<u>ere the valu</u> <u>a</u> Multipur	<u>ie of</u> the <u>char</u> pose Interne
The <u>value h</u> acter string – <u>A 1</u> Mai – <u>The</u>	eld by the Format element s shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital.	hall be a chara lexical repres alue, from RFC	<u>entati</u>	<u>on of</u>	<u>ere the valu</u> <u>a</u> Multipur	<u>ie of</u> the <u>char</u> pose Interne
The <u>value h</u> acter string – <u>A 1</u> Mai – <u>The</u> 5.4.4.2 Siz	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital.	hall be a chara lexical repres alue, from RFC	<u>entati</u>	<u>on of</u>	<u>ere the valu</u> <u>a</u> Multipur	<u>ie of</u> the <u>char</u> pose Interne
The <u>value h</u> acter string - <u>A 1</u> Mai - <u>The</u> 5.4.4.2 Siz The value h	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. ce eld by the Size element shal	hall be a chara lexical repres alue, from RFC	cter stress centati 2 2048 tion o	ning wh	<u>ere the valu</u> <u>a</u> Multipur gits "0"	e of the char pose Interne
The <u>value h</u> acter string - <u>A 1</u> Mai - <u>The</u> 5.4.4.2 Siz The value h value in a I	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. ce eld by the Size element shal <u>COM XML instance shall be</u>	hall be a chara lexical repres alue, from RFC l be a combina a valid non-ne	cter stress contation o contactor of the stress of the str	on of 3: or f the di	<u>a Multipur</u> a Multipur gits "0" r as defined	<u>e of</u> the <u>char</u> pose Interne "9". <u>The Siz</u> I by the XMI
The <u>value h</u> acter string - <u>A 1</u> Mai – <u>The</u> 5.4.4.2 Siz The value h value in a I <u>Schema der</u>	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. coken non-digital. coken non-digital.	hall be a chara lexical repres alue, from RFC l be a combina <u>a valid non-ne</u> EInteger (see	tion o	on of 3: or f the di <u>c intege</u> Schen	a_Multipur a_Multipur gits "0" r as defined na, Part 2).	<u>e of</u> the <u>char</u> pose Interne "9". <u>The Sizu</u> I by the XMI
The value h acter string - <u>A 1</u> Mai - <u>The</u> 5.4.4.2 Siz The value h value in a I Schema der 5.4.4.3 Re	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. ce eld by the Size element shal .OM XML instance shall be ived data type nonNegative quirement	hall be a charac lexical repres alue, from RFC l be a combina a valid non-ne eInteger (see	cter st sentati 2 2048 tion o gative	on of or f the di <u>e intege</u> Schen	a_Multipur a_Multipur gits "0" r as defined na, Part 2).	<u>e of</u> the <u>char</u> pose Interne "9". <u>The Siza</u> I by the XMI
The value h acter string - <u>A 1</u> Mai - <u>The</u> 5.4.4.2 Siz The value h value in a I Schema der 5.4.4.3 Re Table 10 de	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. ce eld by the Size element shal <u>OM XML instance shall be</u> ived data type nonNegative quirement scribes the Requirement elem	hall be a character lexical represent alue, from RFC l be a combina a valid non-ne integer (see	tion o	on of or f the di intege Schen	a_Multipur gits "0" r as defined na, Part 2).	<u>e of the char</u> pose Interne "9". <u>The Siz</u> I by the XMI
The value h acter string - <u>A 1</u> Mai - <u>The</u> 5.4.4.2 Siz The value h value in a I Schema der 5.4.4.3 Re Table 10 de	eld by the Format element si shall be one of the following: iteral that is the canonical l Extension (MIME) type v token non-digital. eld by the Size element shal <u>OM XML instance shall be</u> ived data type nonNegative quirement scribes the Requirement elem Table 10—Th	hall be a characher lexical represent alue, from RFC l be a combina a valid non-ne eInteger (see hent and its dire	tion o cative XMI	f the di cor f the di cor f the di cor cor cor cor cor cor cor cor cor cor	a_Multipur a_Multipur gits "0" r as defined na, Part 2).	<u>e of</u> the <u>char</u> pose Interne "9". <u>The Size</u> I by the XMI

	LOM <u>_data</u> <u>element</u>	XML name	Subelements	Min	_Max_	Order	LOM data type	ŕ
I	Requirement	requirement,	<u>OrComposite</u>	<u>0</u>	= <mark>v</mark> ≊====== (40)	Unordered	Unspecified	1.
	OrComposite	orComposite	<u>See 5.4.4.3.1</u>	0	= <mark>≪</mark> ===== (<u>40)</u>	Unordered	Unspecified	1

544 **5.4.4.3.1 OrComposite**

545 Table 11 describes the OrComposite element and its <u>direct-descendant</u> subelements.

Copyright $\ensuremath{\mathbb{C}}$ 2004 IEEE. All rights reserved.

546

Table 11—The OrComposite element

	LOM, data element	XML name	Subelements	Min	Max	Order	LOM data type		Deleted: name
Î	OrComposite	orComposite	Туре	0	<i>_</i> 00	Unordered	Unspecified	5	Deleted: Data
	I		Name		(40)				Deleted: *
			Minimum Version						Deleted: <
			Maximum Version						Deleted: >
I	Tupa	time	See 5.5.5	0	1	Unepecified	Vocabulary		Deleted: n
	Type	<u>+</u> -y₽c y	Nec 5.5	= <u> </u>	= ¹ = = = =				Deleted: <
Î	Name	name	<u>See 5.5.5</u>	= = = = =		Unspecified	Vocabulary		Deleted: >
	Minimum_Version	minimumVersion	<u>None</u>	0	_1	Unspecified	CharacterString		Deleted: Source¶
	Maximum_Version	maximumversion	None	_0		Unspecified	CharacterString		Deleted: <
547	5.4.4.3.1.1 Type								Deleted: >
548	The Type and Nan	ne elements shall b	e <u>present</u> as a pa	air. If o	ne eler	nent is <u>pres</u>	ent in a LOM		Deleted: Source¶ Value
549	<u>XML</u> instance, the	other shall be prese	<u>ent,</u> also.						Deleted: <
550	If the value space	for the Source s	ubelement of the	Tune	eleme	nt is the I	OMv10 base		Deleted: >
551	schema (i.e. < sou	101 the 500000 solution 101 the 5000000 solution 100000000000000000000000000000000000	sources) then	the va	lid val	les for the	Value subele-		Deleted: <
552	ment of the Type el	lement shall come	from the followin	ng list o	f token	IS:			Deleted: >
553	- operatin	ig system		0					Deleted: *Common data types and data elements are defined in . ¶
554	- browser								Deleted: used
554	- DIOWSEI								Deleted: used
555	5.4.4.3.1.2 Name								Deleted: used
									Deleted: source of the
556	The Type and Nan	ne elements shall t	e <u>present</u> as a pa	ur. It o	ne eler	nent 15 pres	ent in a LOM		Deleted: <\
221	<u>XML</u> instance, the	other shall be prese	ent, also.						Deleted: value
558	If the value space t	for Source subelen	nent of the Name	eleme	ent is th	e LOMv1.0) base schema		Deleted: be one of
559	(i.e., <source/> L0	Mv1.0), and the value	of the	e Value	e subelemer	t of the Type		Deleted: used
560	element is the token	n operating sys	stem, then the v	alid <mark>va</mark>	lues fo	r the Value	subelement of		Deleted: used
561	the Name element	shall <u>come from</u> the	e following list of	token	s:				Deleted: used
560									Deleted: source of the
302	- pc-aos								Deleted: <\
563	- ms-windo	WS							Deleted: Type element's
564	- macos							, N	Deleted: value
565	- unix								Deleted: be one of
566	- multi-os	5							
567	– none								
568	If the value space	for the Source su	ibelement of the	Name	e eleme	ent is the L	OMv1.0 base		Deleted: source of the
569	schema (i.e., <sou:< td=""><td>rce>LOMv1.0<td>ource>), and the</td><td>ne valu</td><td>e <u>of the</u></td><td>e Value sub</td><td>element of the</td><td>- </td><td>Deleted: <\</td></td></sou:<>	rce>LOMv1.0 <td>ource>), and the</td> <td>ne valu</td> <td>e <u>of the</u></td> <td>e Value sub</td> <td>element of the</td> <td>- </td> <td>Deleted: <\</td>	ource>), and the	ne valu	e <u>of the</u>	e Value sub	element of the	- 	Deleted: <\

Copyright © 2004 IEEE. All rights reserved.

Deleted: Type element's

05 November 2004

570	Type element is the token browser, then the valid values for the Value subelement of the	Deleted: value
571	Name element shall <u>come from</u> the following <u>list of</u> tokens:	Deleted: be one of
570		
372	– any	
573	- netscape communicator	
574	- ms-internet explorer	
575	- opera	
576	- amaya	
	ſ	Deleted C / /
577	5.4.5 Educational	Deleted: first-generation

578 Table 12 describes the Educational element and its <u>direct-descendant</u> subelements.

579

Table 12—The Educational element

LOM <u>data</u> <u>element</u>	XMI-name	Subelements	- Min -	- Max-	Order	LOM data type	
Educational	educational	Interactivity_Type	0	₽	<u>Unordered</u>	Unspecified	11-1
		Learning Resource Type		<u>(100)</u>			
		Interactivity Level					
		Semantic Density					1
		Intended End User Role					
		Context					$-\frac{\eta_{I}}{\eta_{I}}$
		Typical Age Range					- 51
		Difficulty					
		Typical Learning Time					
		Description					
		Language					
Interactivity Type	interactivityType	<u>See 5.5.5</u>	<u> </u>	_1	Unspecified	Vocabulary	
Learning Resource Type	learningResourceType.	<u>See 5.5.5</u>	_0	<u>₹</u>	Ordered	Vocabulary	
Interactivity Level	_interactivityLevel	<u>See 5.5.5</u>	<u> </u>	_1	Unspecified	Vocabulary	
Semantic Density	semanticDensity	<u>See 5.5.5</u>	<u> </u>	_1	Unspecified	Vocabulary	
Intended End User Role	intendedEndUserRole.	. <u>See 5.5.5</u>	<u> 0 </u>	<u>≠</u> <u>(10)</u> = = =	Ordered	Vocabulary	1) 1)
Context	context	<u>See 5.5.5</u>	<u> </u>	<u>≠∞</u> = = = = = = = = = = = = = = = = = = =	Unordered	Vocabulary	

Deleted:	first-generation
Deleted:	name
Deleted:	Data
Deleted:	*
Deleted:	<
Deleted:	>
Deleted:	Unspecified
Deleted:	n
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	n
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	n
Deleted:	<
Deleted:	>
Deleted: Value	Source¶
Deleted:	n

Copyright $\ensuremath{\mathbb{C}}$ 2004 IEEE. All rights reserved.

LOM <u>data</u> <u>element</u>	XML name	Subelements	- Min -	- Max-	Order	LOM data type,
Typical Age Range	typicalAgeRange	<u>See 5.5.4</u>	<u> </u>	<u>≠∞</u> (5)	<u>Unordered</u>	LangString
Difficulty	difficulty	<u>See 5.5.5</u>	<u>0</u>	_1	Unspecified	Vocabulary
Typical Learning Time	typicalLearningTime,	<u>See 5.5.3</u>	_0	_1	Unspecified	
Description	description	<u>See 5.5.4</u>	<u> </u>	<u>≠∞</u>	<u>Unordered</u>	LangString
Language	-language	None	_0	<u>≁∞</u> (<u>10)</u> = =	Unordered	CharacterString_

580 **5.4.5.1 Interactivity Type**

If the value space for the <u>Source subelement of the</u> Interactivity Type element is the LOMv1.0
base schema (i.e., <source>LOMv1.0
subelement of the Interactivity Type element shall come from the following list of tokens:

- 584 active
- 585 expositive
- 586 mixed

587 **5.4.5.2 Learning Resource Type**

If the value space for the <u>Source subelement of the Learning</u> Resource Type element is the
LOMv1.0 <u>base</u> schema (i.e., <<u>source>LOMv1.0</u>
<u>Value subelement</u> of the <u>Learning Resource Type element shall come from the following list</u>
of tokens:

- 592 exercise
- 593 simulation
- 594 questionnaire
- 595 diagram
- 596 figure
- 597 graph
- 598 index
- 599 slide
- 600 table
- 601 narrative text
- 602 exam
- 603 experiment

Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: String
Deleted: Unspecified
Deleted: n
Deleted: <
Deleted: >
Deleted: Source¶ Value
Deleted: <
Deleted: >
Deleted: Duration¶ Description
Deleted: <
Deleted: >
Deleted: String
Deleted: Unspecified
Deleted: n
Deleted: <
Deleted: >
Deleted: n
Deleted: *Common data types and data elements are defined in . ¶
Deleted: source of the
Deleted: <\
Deleted: value
Deleted: element shall be one
Deleted: source of the
Deleted: bas
Deleted: <\
Deleted: value
Deleted: element shall be one

Draft P1484.12.3/D5

- 604 problem statement
- 605 self assessment
- 606 lecture

607 **5.4.5.3 Interactivity Level**

608 If the value space for the <u>Source subelement of the</u> Interactivity Level element is the 609 LOMv1.0 base schema (i.e., <source>LOMv1.0</source>), then the valid <u>values</u> for the 610 <u>Value subelement</u> of the <u>Interactivity Level element shall come from the</u> following list of to-611 kens:

- 612 very low
- 613 low
- 614 medium
- 615 high
- 616 very high

617 5.4.5.4 Semantic Density

618 If the value space for the <u>Source subelement of the Semantic Density element is the LOMv1.0</u> 619 base schema (i.e., <source>LOMv1.0620 subelement of the Semantic Density element shall come from the following list of tokens:

- 621 very low
- 622 low
- 623 medium
- 624 high
- 625 very high

626 5.4.5.5 Intended End User Role

If the value space for the <u>Source subelement of the Intended End User Role element is the</u>
LOMv1.0 base schema (i.e., <source>LOMv1.0
Value subelement of the Intended End User Role element shall come from the following list
of tokens:

- 631 teacher
- 632 author
- 633 learner
- 634 manager

-1	Deleted: source of the
- 1	Deleted: <\
-+	Deleted: value
-+	Deleted: element shall be one

1	Deleted: source of the
1	Deleted: <\
1	Deleted: value
1	Deleted: element shall be one

1	Deleted: source of the
-{	Deleted: <\
-{	Deleted: value
-	Deleted: element shall be one

635 5.4.5.6 Context

636 If the value space for the <u>Source subelement of the Context element is the LOMv1.0 base</u>
637 schema (i.e., <source>LOMv1.0
638 ment of the <u>Context</u> element shall <u>come from</u> the following <u>list of</u> tokens:

- 639 school
- 640 higher education
- 641 training
- 642 other

643 **5.4.5.7 Difficulty**

644 If the value space for the <u>Source subelement of the</u> Difficulty element is the LOMv1.0 base
645 schema (i.e., <source>LOMv1.0
646 ment of the Difficulty element shall <u>come from</u> the following <u>list of</u> tokens:

- 647 very easy
- 648 easy
- 649 medium
- 650 difficult
- 651 very difficult

652 **5.4.5.8 Language**

The Language element shall be a character string where the value of the character string shall have the format and value space described in 5.5.4.1.

655 5.4.6 Rights

656

657

Table 13—The Rights element

Table 13 describes the Rights element and its direct-descendant subelements.

	LOM_ <u>data</u> <u>element</u>	XML-name	-Subelements	Min	-Max-	Order	<mark>LOM data</mark>	/
l	Rights	_rights	Cost	0	1	Unspecified_	Unspecified	<
I			Copyright And Other Restrictions					· · · · ·
			Description					
	Cost	cost	<u>See 5.5.5</u>	0	1	Unspecified	Vocabulary	<u>[</u>

1	Deleted: source of the
-{	Deleted: <\
-	Deleted: value
ſ	Deleted: be one of

Deleted: source of the

Deleted: <\

Deleted: value Deleted: be one of

. {	Deleted: be
-{	Deleted: values
` {	Deleted: 5.51
.{	Deleted: first-generation

Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: and
Deleted: <
Deleted: >
Deleted: Source¶ Value

Relation

Kind

674

675 676

677

Resource

5.4.7.1 Kind

relation

kind

resource

Kind___

Resource

See 5.5.5

See 5.4.7.2

<u>Kind</u> element shall <u>come from</u> the following <u>list of</u> tokens:

Deleted: name Deleted: Data Deleted: * Deleted: < Deleted: > Deleted: Source¶

Deleted: <

Deleted: String

Deleted: value
Deleted: be one of

elements are defined in . ¶

Deleted: source of the Deleted: <\

Deleted: source of the

Deleted: Copy Right

Deleted: element shall be one

Deleted: first-generation
Deleted: name

Deleted: <\

Deleted: value

Deleted: *Common data types and data

	LOM <u>data</u> <u>element</u>	XML-name		ements - Min	Max-	Order	<mark>LOM data</mark>
	Copyright And Other Restrictions	¢opyrightAndOtherRes	trictions <u>See 5.</u>	<u>550_</u>		_Unspecified_	Vocabulary
ļ	Description	description	<u>See 5.</u> :	5.40	1	Unspecified	LangString
3	5.4.6.1 C	ost					
)	If the value	e space for the Source	e subelement of the	Cost elem	ent is th	e LOMv1.0	base schema
)	(1.e., <sou Cost eleme</sou 	rce>LOMv1.0 source	urce>), then the ne following list of	alid values	for the	Value sube	lement of the
	_ V9	.s					
	- no						
	If the value is the LOI for the <u>Va</u> <u>the</u> following	e space for the <u>Source</u> Mv1.0 base schema (<u>lue subelement</u> of the ing <u>list</u> tokens:	e subelement of th (i.e., <source/> LC e <u>Copyright And (</u>	e Copyright Mv1.0Other Restr	And Cources	ther Restric >), then the element sha	tions element valid <u>values</u> ll come from
)	– уе	S					
)	– no						
	5.4.7 R	elation					
2	Table 14 d	escribes the Relation	element and its dir	ect-descend	<mark>lant</mark> sub	elements.	
· ·		Tat	ole 14—The Rel	ation elen	nent		

14	Deleted: Data
	Deleted: *
- 111	Deleted: <
11/1	Deleted: >
11/1	Deleted: n
11/1	Deleted: <
11 11	Deleted: >
	Deleted: Source¶ Value
	Deleted: <
1	Deleted: >
	Deleted: Identifier¶ Description
	Deleted: *Common data types and data elements are defined in . ¶
	Deleted: source of the
	Deleted: <\
	Deleted: value
	Deleted: be one of

Copyright © 2004 IEEE. All rights reserved.

If the value space for the Source subelement of the Kind element is the LOMv1.0 base schema

(i.e., <source>LOMv1.0</source>), then the valid values for the Value subelement of the

0__

0_

0

<u>(100)</u>

1___

1

Unordered

Unspecified

Unspecified

Unspecified

Vocabulary_

Unspecified

678	-	ispartof

- 679 haspart _
- 680 isversionof _
- 681 hasversion _
- 682 isformatof
- 683 _ hasformat
- 684 references _
- 685 isreferencedby _
- 686 isbasedon _
- 687 isbasisfor _
- 688 requires _
- 689 isrequiredby _

690 5.4.7.2 Resource

691

Table 15 describes the Resource element and its direct-descendant subelements.

692

Table 15—The Resource element

LOM <u>data</u>	XML- name	Subelements	Min	Max	Order	LOM data type
Resource	resource	_Identifier	0	_1	Unspecified	Unspecified
		Description				
Identifier	identifier	See 5.4.7.2.1	0	= <u>*</u> ===================================	<u>Unordered</u>	
Description	description	<u>See 5.5.4</u>	0	= <u>∞</u> (10)	<u>Unordered</u>	LangString

693 5.4.7.2.1 Identifier

694 Table 16 describes the Identifier element and its direct-descendant subelements.

695

Table 16—The Identifier element

Table 16—The Identifier element								
LOM <u>data</u> <u>element</u>	XML-name	Subelements	-Min-	Max-	Order	- <u>LOM-data</u> typę, -		
Identifier	identifier	<u>Catalog</u>	0	* ====	Unordered	Unspecified		
		Entry		<u>(10)</u>				
Catalog	catalog	<u>None</u>	0	1	Unspecified	_CharacterString	2	
Entry	entry	None	0	_1	Unspecified	CharacterString		

Deleted: first-generation
Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: <
Deleted: >
Deleted: Catalog¶ Entry
Deleted: Unspecified
Deleted: n
Deleted: <
Deleted: >
Deleted: String
Deleted: Unspecified
Deleted: n
Deleted: *Common data types and data elements are defined in . ¶
Deleted: first-generation
Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: Unspecified
Deleted: n
Deleted: <
Deleted: >
Deleted: <
Deleted: >

696 **5.4.8** Annotation

697

Table 17 describes the Annotation element and its direct-descendant subelements.

698

Table 17—The Annotation element

LOM <u>data</u> <u>element</u>	XML-name	Subelements	Min	Max	Order	LOM data type	
Annotation	annotation	Entity	0	- * ====	Unordered	Unspecified	4
		Date		<u>(30)</u>			
		Description					
Entity	entity	<u>None</u>	0	1	Unspecified	CharacterString	1
Date	date	<u>See 5.5.2</u>	0	1	Unspecified	<u>DateTime</u>	_
Description	description	<u>See 5.5.4</u>	0	1	Unspecified	LangString	

699 **5.4.8.1 <u>Entity</u>**

700The value held by the Entity element shall be a character string literal that is the canonical701lexical representation of a valid vCard as defined in IETF RFC 2426:1998.

702NOTE—IETF RFC 2426:1998 does not rely on XML syntax to express the internal structure of a
valid vCard. At the time of publication of this Standard, no *de facto* standard W3C XML Schema

704 definition language binding for the vCard specification existed.

705 5.4.9 Classification

707

706

Table 18 describes the Classification element and its <u>direct-descendant</u> subelements.

LOM <u>,data</u> <u>element</u>	– – – XML- name – – –	– – Subelements – –	- M i n-	- Max -	Order	<mark>ŁOM-data</mark> typę
Classification	classification	Purpose	0	<u></u>	Unordered	Unspecified
		Taxon Path		<u>(40)</u>		
		Description				
		Keyword				
Purpose	purpose	_None	_0	1	_Unspecified _	_Vocabulary
Taxon Path	taxonPath	<u>See 5.4.9.2</u>	0	₽	Unordered	Unspecified
				<u>(15)</u>		
Description	description	<u>See 5.5.4</u>	<u>0</u>	1	Unspecified	_LangString
Keyword	keyword	See 5.5.4	0	<u></u>	Ordered	LangString
-				(40)		

Table 18—The Classification element

Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation

Deleted: name Deleted: Data Deleted: * Deleted: < Deleted: > Deleted: n Deleted: < Deleted: > Deleted: vCard Deleted: < Deleted: > Deleted: DateTime¶ Description Deleted: < Deleted: > Deleted: String Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name **Deleted**: Data Deleted: * Deleted: < Deleted: > Deleted: n Deleted: < Deleted: > Deleted: < Deleted: > Deleted: Source Taxon Deleted: Undefined Deleted: n Deleted: < Deleted: > Deleted: String Deleted: < Deleted: > Deleted: String Deleted: n

Draft P1484.12.3/D5

708 **5.4.9.1** Purpose

709 If the value space for the <u>Source subelement of the Purpose element is the LOMv1.0 base</u>
710 schema (i.e., <source>LOMv1.0
711 ment of the Purpose element shall come from the following list of tokens:

- 712 discipline
- 713 idea
- 714 prerequisite
- 715 educational objective
- 716 accessibility restrictions
- 717 educational level
- 718 skill level
- 719 security level
- 720 competency

721 **5.4.9.2 Taxon Path**

722

723

Table 19—The Taxon Path element

Table 19 describes the Taxon Path element and its direct-descendant subelements.

LOM <u>data</u>	XMI- name	Subelements	Min	Max	Order	LOM data type
Taxon Path	taxonPath	Source	0	- <u>x</u> (<u>15)</u>	Ordered	Unspecified
Source	source	<u>See 5.5.4</u>	0	1	Unspecified	LangString
Taxon	taxon	<u>See 5.4.9.2.1</u>	<u>0</u>	<u></u>	Ordered	Unspecified

724 5.4.9.2.1 Taxon

Table 20 describes the Taxon element and its <u>direct-descendant</u> subelements.

726

Table 20—The Taxon element

LOM <u>data</u>		Subelements	Min	Max	Order	LOM data type,	
Taxon	taxon	Id	0	<u>~</u>	Ordered	Unspecified	",
		Entry		<u>(15)</u>			
Id	įd	<u>None</u>	0	1	Unspecified	CharacterString	į
Entry	entry	<u>See 5.5.4</u>	0	1	Unspecified	LangString	12

1	Deleted: Common data types and data elements are defined in . ¶
1	Deleted: source of the
1	Deleted: <\
1	Deleted: value
4	Deleted: be one of

Deleted: first-generation
Deleted: name
Deleted: Data
Deleted: *
Deleted: <
Deleted: >
Deleted: n
Deleted: <
Deleted: >
Deleted: String
Deleted: <
Deleted: >
Deleted: Id¶ Entry
Deleted: n
Deleted: *Common data types and data elements are defined in . ¶
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: *
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: <
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: <
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: <
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: <
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: > Deleted: > Deleted: n Deleted: n Deleted: >
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: Data Deleted: * Deleted: > Deleted: > Deleted: n Deleted: > Deleted: <
Deleted: *Common data types and data elements are defined in . ¶ Deleted: first-generation Deleted: name Deleted: > Deleted: <

727

5.5 Common data types and elements

728 Data types that are used in the LOM-XML <u>Schema</u> binding are defined in 5.5.1 = 5.5.5.

729 5.5.1 CharacterString

- 730 For those elements defined in this <u>Standard</u> that have a <u>LOM</u> data type of CharacterString, the
- W3C XML Schema definition language binding shall <u>have an XML Schema primitive data</u>
 type of string. The string data type supports the repertoire of ISO/IEC 10646–1. The
- 733 string data type ensures the support of multiple languages including multibyte languages.
- NOTE—The XML Schema definition language primitive data type string is defined in "XML
 Schema Part 2."

736 **5.5.2 DateTime**

- The DateTime data type is represented as a set of XML elements. LOM data elements with a
 data type of DateTime may contain the following subelements:
- 739 <u>DateTime</u>
- 740 <u>Description</u>
- 741 If present, the subelements listed above shall be contained by the element for which the
 742 DateTime data type is defined. The order of appearance of subelements shall not be signifi743 cant.
- 744 <u>Table 21 describes the DateTime data type and its direct-descendant subelements.</u>
- 745

Table 21—The DateTime data type

<u>LOM data</u> <u>element</u>	XML name	<u>Subelements</u>	<u>Min</u>	<u>Max</u>	<u>Order</u>	LOM data type
DateTime		DateTime	<u>0</u>	1	Unspecified	Unspecified
		Description				
DateTime	dateTime	None	<u>0</u>	<u>1</u>	Unspecified	CharacterString
Description	description	<u>See</u> 5.5.3.2	<u>0</u>	<u>1</u>	Unspecified	LangString

746 <u>NOTE—The DateTime data type is made up of two elements, DateTime and Description, that are</u>

747 748 direct descendents of all aggregate elements that are defined to be of type DateTime. For example, DateTime and Description are the direct descendents of the aggregate element Date (see

749 5.4.2.2).

750 **5.5.2.1 DateTime**

751 The DateTime element is a pattern that shall be defined according to the following constraints;

Copyright © 2004 IEEE. All rights reserved.

27

Deleted: *Common data types and data elements are defined in . ¶ Deleted: 1.1.1

Deleted: standard	
Deleted: be	
Deleted: xs:	
Deleted: xs:	
Deleted: -	
Deleted: xs:	
Deleted: <xs:< td=""><td></td></xs:<>	
Deleted: > construct	_

XML element. The DateTime data type may contain the following subelements:¶ <#>Date Time¶ <#>Description¶ If present, the subelements listed above shall be contained by the element for which the DateTime type is defined. The order of appearance of subelements shall not be significant.¶ **<#>Date Time¶** Table 21 describes the Date Time element and its first-generation subelements.¶ **Table 21—The DateTime element¶** LOM name[28]

Deleted: e data type is represented as an

05 November 2004

752	YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]] where:	
753	- yyyy: The 4-digit year (>0001)	Deleted: is the four
755		Deleted: (>=
754	 MM: The 2-digit month (01 through 12 where 01=January, etc.) 	Deleted: is the two
755 756	 DD<u>; The 2</u>-digit day of month (01 through 31, depending on the values of month and year) 	Deleted: is the two
757	- bb [:] Two digits of hour (00 through 23) (am and pm are NOT allowed)	Deleted: is two
750	True divite of minute (00 through 20) (an <u>ance</u> pin <u>ano</u> r(01 anonea)	Deleted: /
/58	- mm <u>, 1wo</u> digits of minute (00 through 59)	Deleted: is two
759	 ss<u>; Two</u> digits of second (00 through 59) 	Deleted: is two
760	 s: <u>One</u> or more digits representing a decimal fraction of a second 	Deleted: is one
761	- TZD: The time zone designator (z for coordinated universal time [UTC] or +hh,	Deleted: is the
762	-hh, +hh:mm, or -hh:mm, where hh is two digits of hour and mm is two digits of	Deleted: "
763	<u>minute</u>)	Deleted: "
764	If the DateTime element contains a value, at least the four-digit year shall be present. If addi-	Deleted: At
765	tional parts of the date and time are included, the character literals "-", T, ":", and "." are part	Deleted: "
766	of the character lexical representation for the <u>DateTime element</u> .	Deleted: "
767 768	If the time portion is present, but the time zone designator is not present, the time zone is in- terpreted as being UTC.	Deleted: datetime
769	NOTES:	
770 771	1—Because of restrictions placed on the value space, the W3C XML Schema definition language	Deleted: vs:
772	Therefore, a regular expression was created to represent all valid dates and times supported by	Deleted: in the XSD files provided with
113	IEEE 1484.12.1–2002.	Deleted: capture
774	2-The date portion represents dates in the Common Era (CE), only. The date portion follows the	Deleted: /
775	Gregorian calendar for dates from October 15, 1582, forward, and the Julian calendar for dates	Deleted:)
777	cases should be represented using the Description element (see 5.5.2).	Deleted: after
9770	2 The approximation there are a share atoms (f and 1) in directs and in the stars at the stars are for	Deleted: ("["
779	5— The square bracket meta characters [1] and [1] indicate optional elements that may appear zero	Deleted: "]")
780	appear in the result; only the associated values described appear (e.g., DD is replaced by the corre-	Deleted: "
781	sponding 2-digit value for day of month).	Deleted: "

- 783 **5.5.2.1.1 Description**
- The Duration data type is represented as a set of XML elements. LOM data elements with a
 data type of Duration may contain the following subelements:

4—This value space is based on ISO 8601:2000 [B2].

786 – <u>Duration</u>

782

Copyright © 2004 IEEE. All rights reserved.

Deleted:

Deleted: A3

Deleted:

Deleted: element
Deleted: first-generation

Deleted: element Deleted: name

Deleted: Data
Deleted: unspecified

Deleted: < Deleted: > Deleted: < Deleted: String Deleted: which Deleted: rules Deleted: is the

Deleted: >

Deleted: > Deleted: is the

Deleted: > Deleted: is the Deleted: > Deleted: is the Deleted: > Deleted: is the Deleted: > Deleted: ' Deleted: ", " Deleted: ' Deleted: " Deleted: "

Deleted: is the

Deleted: , not restricted

Deleted: and value must always be present. The designator "P" is always

Deleted: . The

787 – <u>Description</u>

If present, the subelements listed above shall be contained by the element for which the Dura tion data type is defined. The order of appearance of subelements shall not be significant.

790 **5.5.3 Duration**

791 Table 22 describes the Duration data type and its direct-descendant subelements.

792

Table 22—The Duration data type

LOM <u>data</u> <u>element</u>	XML-name	Subelements	Min	Max	Order	LOM data type
Duration	.	Duration	0	1	Unspecified	Unspecified
		Description				
Duration	duration	<u>None</u>	0	1	<u>Unspecified</u>	CharacterString
Description	description	<u>See 5.5.3.2</u>	0	1	Unspecified	LangString

NOTE—The Duration data type is made up of two elements, Duration and Description, that are direct descendents of all aggregate elements that are defined to be of type Duration. For example, Duration and Description are the direct descendents of the aggregate element Duration (see 5.4.4).

797 5.5.3.1 Duration

798 799	The value space for the Duration element is a pattern <u>that</u> shall be defined according to the following <u>constraints</u> :									
800	P[yY][mM][dD][T[hH][nM][s[.s]S]] where:									
801	$-$ y: The number of years (integer, ≥ 0)									
802	- m: The number of months (integer, ≥ 0 , not restricted, e.g., > 12 is acceptable)									
803	- d: The number of days (integer, ≥ 0 , not restricted, e.g., > 31 is acceptable)									
804	- h: The number of hours (integer, ≥ 0 , not restricted, e.g., > 23 is acceptable)									
805	- n: The number of minutes (integer, ≥ 0 , not restricted, e.g., > 59 is acceptable)									
806 807	 s: <u>The number of seconds or fraction of seconds (integer, ≥ 0, not restricted, e.g., /</u> > 59 is acceptable) 									
808 809	The character literal designators P. Y. M. D. T. H. M. and S. shall appear if the corresponding nonzero value is present.									
810 811 812	If the <u>Duration element contains a value</u> , the designator <u>P</u> shall be present. If the value of years, months, days, hours, minutes or seconds is zero, the value and corresponding designation (e.g., <u>M</u>) may be omitted, but at least one designator <u>with a positive value shall be present</u> .									

813	in addition to the designator p . The designator T shall be omitted if all of the time (hours, min-	Deleted: "
814	utes, and seconds) is zero.	Deleted: "
015		Deleted: /
815	Negative durations are not supported.	Deleted: /
816	NOTES:	Deleted: are
017		
817	I—Because of the restrictions placed on the value space, the w3C XML Schema definition lan- guage primitive data type date Time could not be used to create conforming LOM XML in-	
819 820	stances. A regular expression was built to represent all valid date/times supported by IEEE 1484,12,1–2002.	Deleted: AST Deleted: in the XSD files provided with this standard (see Annex B)
		Deleted: capture
821	2—The value is designated in the Gregorian calendar.	
822	3—The ordering of durations may be indeterminate (e.g., 1 month may be 28, 29, 30, or 31 days).	
823 824 825	4— <u>The square bracket meta characters ([and])</u> indicate optional elements that may appear zero or one time in the character lexical representation of the Duration. These meta characters do not appear in the result; only the associated values described appear (e.g., dD is replaced by the corre-	Deleted: For durations that apply only while the learning object is in use, but not when its use is suspended, it is recommended that only hours
826 827	sponding value for the number of days in the duration and is followed by the character literal des- ignator p).	Deleted: smaller units of duration be used. <i>Examples:</i> PT43H, PT5M35S. For durations that express a time span, re-
828	5—This value space is based on ISO 8601:2000 [B2].	gardless of whether the learning object is actually used continuously during that time, days and larger units of duration may be used Examples: PLVM P20D
829	5.5.3.2 Description	5—The square bracket meta characters
820	The Degewight i on data type is represented as a Length wing data type (see 5.5.4)	Deleted: "
830	The Description data type is represented as a Langstring data type (see $\frac{2.3.4}{2.3.4}$).	Deleted: "
		Deleted: "
831	5.5.4 LangString	Deleted: "
832	The LangString data type is represented as a set of XML elements LOM data elements with a	Deleted: 6—
833	data type of LangString may contain the following subelements:	Deleted: A3
000	and type of Lungouring indy contain the following outperformance.	Deleted: 5.5.
834	– <u>String</u>	Deleted: an
835	– Language	Deleted: . Table 23 describes the .
836	If present, the subelements listed above shall be contained by the element for which the	
837	LangString data type is defined. The order of appearance of subelements shall not be signifi-	
838	cant. Within a LOM XML instance, the LOM data element Language shall be specified as an	
839	attribute of LOM data elements that are of type LangString.	Deleted: first-generation
840	Table 23 describes the LangString data type and its direct-descendant subelements.	

Deleted: element

841

Table 23—The LangString data type

	· · · · · · · · · · · · · · · · · · ·									
	LOM,data	XML - name	Subelements	Min	Max	Order	LOM data-type -		Delet	ted: name
	element					01401			Delet	ted: Data
	LangString	₹	String	0	-	Unordered	Unspecified		Delet	ted: Unspecified
			Language		<u>(10)</u>				Delet	ted: n
	String	string	None	0	_1	Unspecified	CharacterString		Delet	ted: <
	Language	language	_None	0	_1	Unspecified	CharacterString		Delet	ted: >
842 843 844	<u>NOTE—The I</u> rect descenden String and Lan	angString data ty ts of all aggregate guage are the dire	be is made up of two elements that are do the descendents of the	o elements efined to b e aggregat	s <u>, String</u> e of type e elemer	and Langua LangString t Title (see 1	ge, that are di- . For example, 5.4.1 <u>).</u>		Delet Ute, no The L ter str	ted: * ted: *language is an XML attrib- ot an XML element.¶ .anguage element shall be a charac- ing consisting of a required lan- code followed by multiple
845 846	5.5.4.1 Lang	guage	a character string	consisting	t of a ra	mired lang	unga coda fol		optior examp is defi	ial, hyphen-prefixed subcodes (see ples below). The Language element ined in
840 847	lowed by mult	tiple optional by	ben-prefixed subc	odes (Consisting		<u>quired lange</u>	lage code loi-	T7	Delet	ted: ¶
848 849	The following constraints apply to the language code part of the character string:									ted: valid LanguageID where¶ ruageID = Langcode ("- bcode)*¶ blowing rules apply to the Lang- e part of the
850	2 lotte	r aadas ara dafin	ad in ISO 630 2						Delet	ted: :
830	- 2-10110	er coues are defin	ieu ili 130 039–2.						Delet	ted: Two
851	$-$ The $\frac{1}{2}$	<u>-letter code</u> "i" i	s reserved and use	<u>es as a pr</u>	<u>efix</u> for	registration	ns defined by	`	Delet	ted: Three
852	the In	ternet Assigned I	Numbers Authority	(IANA)					Delet	ted: value
853	- The 1	<u>-letter code</u> "x" is	s reserved and use	<u>d as a pre</u>	<u>fix</u> for p	rivate use.			Delet	ted: value
854	The following	constraints apply	to the first subcode	part of th	ne charac	ter string:		1	Delet	ted: rules
0.5.5		1 1 1		·····	1				Delet	led: Subcode
833	- 2 -lette	er subcodes are is	SO 31661 alpha-2	country c	odes.				Delet	ted: 1 wo
856	– Subco	odes from 3 to 8 i	<u>n length</u> letters are	registere	ed with l	ANA.			Delet	ted: 01
857	<u>Constraints</u> fo	r additional subco	des are unspecified	:				1	Delei	ieu. Ruics
858	The value hel	d by the character	r string shall be a v	alid lanoi	age cod	e as defined	by the XML	1	Delet	t ed: NOTE—The Lang
859	Schema derive	ed data type lang	uage (see XML So	chema, Pa	<u></u>					
860 861	ISO 639–2 sp for terminolog	ecifies two code s sy applications (IS	sets, one for bibliog O 639–2/T). Either	raphic ap	plicatior may be	<u>is (ISO 639- used.</u>	-2/B) and one			
862	NOTES:									
863 864	<u>1—The langua</u> However, the v	nge_code is norma values are case inse	lly given in lower c ensitive.	ase and th	ne subco	des (if any)	in upper case.			
865 866	<u>2—The XML</u> language code	Schema derived of that are described	lata type language above.	e does not	t enforce	all the con	straints on the			

Copyright © 2004 IEEE. All rights reserved.

31

867	Examples									
868 869 870 871	"en-GB" "de" "fr-CA" "it"									
872	"i-bnn"	(IANA Bunun)						11	Deleted: en-US-philadelphia"	
873	5.5.5 Voc	abulary		Deleted: <#>Token¶ For those elements defined in this stan- dard that have a data type of Token, the W3C XML Schema definition language						
874 875	<u>The Vocabula</u> data type of V	ry data type is rej ocabulary may co	- 1	binding shall be an xs:token. The xs:token data type supports the reper- toire of ISO/IEC 10646-1. The xs:token data type ensures the support						
876	– Sourc	e							of multiple languages including multibyte	
877	– Value	2							NOTE—The <xs:token> construct is defined in "XML Schema Part 2."¶ <#>vCard¶</xs:token>	
878 879	If present, the cabulary data	subelements list type is defined. T	ed above shal he order of ar	ll be contair opearance of	ned by the e f subelement	lement for y is shall not b	which the Vo- be significant.		The vCard data type is represented as a CharacterString data type. The canonical lexical representation of the vCard uplue shell be a valid vCardI as	
880	Table 23 desc	ribes the Vocabul	ary data type	and its direc	ct-descendar	it subelemer	<u>nts.</u>		defined in IETF RFC 2426:1998.¶ NOTE—IETF RFC 2426:1998 does not rely on XML syntax to express the inter-	
881		<u>Table</u>	e 24—The V	ocabulary	y data type	2			nal structure of a valid vCard[]. At the time of publication of this standard, no <i>de</i> <i>facto</i> standard W3C XML Schema defini-	
	LOM <u>data</u> <u>element</u>	XML-name	Subelements	Min	Max	Order	LOM data type		tion language binding for the vCard specification existed.¶	
ļ	Vocabulary	¥	Source Value	Unspecified	Unspecified	Unspecified	Unspecified		Deleted: The vocabulary data type is represented as an XML element. Table 24 describes the Vocabulary element and its first-generation subelements.¶	
	Source	source	_None	<u>0</u>	1	Unspecified	CharacterString		Table 23—The Vocabulary element¶	
	Value	value	_None	0	1	Unspecified	CharacterString		Deleted: Data	
882	NOTE—The	Vocabulary data ty	pe is made up	o of two eler	nents, Sourc	e and Value,	that are direct		Deleted: Unspecified	
883	descendents o	f all aggregate ele	ements that ar	re defined to	be of type	Vocabulary.	For example,		Deleted: <	
884 885	Source and V 5 4 1 4	alue are the direc	t descendents	of the aggr	regate eleme	nt Aggregat	ion Level (see	- 11 11	Deleted: >	
005	J.T.1.T <u>J.</u>							11 11	Deleted: <	
886	5 5 5 1 Valu							N N	Deleted: >	
000	0.0.0.1 <u>Val</u>								Deleted: Token	
887 888 889	The value held by the character string shall be a valid token as defined by the XML Schema derived data type token (see XML Schema, Part 2). Valid values for the tokens are defined for those LOM data elements that are of type Vocabulary (see 5.4).									
890 891	for those LOM data elements that are of type Vocabulary (see 5.4). NOTE—The XML Schema derived data type token supports the repertoire of ISO/IEC 10646— 1. The token data type ensures the support of multiple languages including multibyte languages. Multibute languages are not used by the LOMul 0 have scheme useshularies but may be used if									

893 the source of a vocabulary is not LOMv1.0.

894 Annex A

895 (informative)

896 Bibliography

897 898	[B1] IEEE 100 [™] , The Authoritative Dictionary of IEEE Standards Terms, Seventh Edi-	Deleted: A1
899 900	[B2] ISO 8601:2000, Data elements and interchange formats – Information interchange – Representation of dates and times.	Deleted: [A2] IETF RFC 2425:1998, MIME Content-Type for Directory In- formation.¶
901	[B3] ISO/IEC 11404:1996, Information Technology—Programming Languages, their Envi-	Deleted: A3 Deleted: A4
902	ronments and System Software Interfaces—Language-Independent Datatypes.	Deleted: [A5] W3C Recommendation
903	[B4] W3C Recommendation (04 February 2004), Extensible Markup Language (XML) 1.1.	24 October 2001, XML Information Set.¶
904	[B5] W3C Recommendation (04 February 2004), XML Information Set (Second Edition).	

905 Annex B

906 (informative)

907 Internet availability of XSD files

908This Standard includes XSD files that are available on the World Wide Web but are not in-
cluded in the printed version of this Standard. These XSD files are available for direct inclu-
sion in applications and for use as examples for those who prefer to develop their own XSDs.

- 911 The XSD files are available at the following URL:
- 912 http://ltsc.ieee.org/xsd/lomv1.0/
- 913 NOTE—This Standard does not require the use of an XSD to validate a LOM XML instance.

-1	Deleted: standard
- 1	Deleted: standard.
- 1	Deleted: as examples
- 1	Deleted: XSD files

914 Annex C

915 (informative)

916 **XSD file descriptions**

917 918 919	The LOMv1.0 base schema describes a structured collection of standard data items, including their data types, multiplicities, and container/component relationships, but does not provide a syntax for encoding LOM data that conforms to the LOMv1.0 base schema.		Deleted: concrete
920 921 922	The LOM XML binding described in this standard defines the XML syntax for encoding LOM data. The binding is a collection of rules describing how to express LOM data in XML syntax. These binding rules are defined in detail in Clause 5 of this standard.		
923 924	When building XSDs, implementers can decide when to enforce LOMv1.0 base schema con- straints. These constraints can be defined in an XSD with the intent that schema processors		Deleted: where
925	will perform validation of the constraints or that validation will be performed by processing.		Deleted: defined in the XSD,
926	Clause 5 defines the criteria to create valid LOM instances; XSD implementers must decide		Deleted: outside the XSD
927 928	how to develop XSD schema files according to these constraints. Regardless of the implemen- tation, a LOM instance must conform to the rules defined in Clause 5 <u>PSVI</u> .		Deleted: in some implementation- defined, post-schema
929	This Standard includes informative W3C XML XSDs for implementers to satisfy many of the		Deleted: standard
930	requirements of IEEE 1484 12 1–2002 (see Annex B) and the practices of its existing adopt-		
931	ers. The XSDs are intended to be used by XML processing tools that support the W3C XML		Deleted: (see Annex B)
932	Schema definition language (see XML Schema, Parts 1 and 2). The LOMv1.0 base schema		
933	places requirements on a LOM XML instance some of which cannot be expressed in W3C		
934	XML Schema definition language. These requirements are defined in Clause 5 of this Stan-		
935	dard and the LOMv1.0 base schema.		
936	The goal is to provide alternative XSDs, each of which supports the LOM XML Schema bind-		Deleted: A predefined, composite XSD (see C.1.1) is provided that can be tai-
937	ing by making certain choices about which binding constraints to describe in the		lored in multiple ways for particular
938	XSDs. <u>Constraints</u> defined in an XSD should be enforced by tools that support XML Schema		checking vocabulary values, and using
939	validation. <u>Constraints</u> not expressed in an XSD are enforced by other means, such as	N N	extensions. Three additional predefined
940	processing the PSVI. No matter where <u>constraint</u> enforcement is <u>performed</u> , a <i>conforming</i> or	11	force different validation rules (see C.1.2
941	strictly conforming LOM XML instance has to conform to all constraints defined in Clause 5		 C.1.4). The four composite XSDs draw from several sets of component XSDs
942	of this Standard and the LOMv1.0 base schema.		(see C.2).¶
0/3	A predefined composite XSD (see $C(1,1)$ is provided that can be tailored in multiple ways for		Deleted: rules
945 944	A predenied, composite ASD (see C.1.1) is provided that can be tailored in multiple ways for particular preferences about LOM data element ordering, vocabulary values checking, and	1.1.1	Deleted: Rules
945	extension use Three additional predefined composite XSDs are provided that enforce differ-		Deleted: Rules
946	ent validation constraints (see C.1.2 – C.1.3). The four composite XSDs draw from several	$-\frac{1}{1}$	Deleted: the rule
947	sets of component XSDs (see C.2).		Deleted: done
		- Y	Deleted: must
948	The need for alternative XSDs was motivated by the variety of ways in which application de-	. i	Deleted: rules

948 The need for alternative XSDs was motivated by the variety of ways in which application de-949 velopers need to use the XSDs. The approach of providing alternative XSDs addresses the



Deleted: composite XSD

i			
976	C.1.1 IomStrict.xsd		Deleted: <#>Does not provide valida- tion to any defined vocabulary schema (i.e., the LOMv1.0 base schema or a
977	The lomStrict.xsd composite XSD encodes a particular set of binding constraints in the		custom schema). A validating parser will
978	W3C XML Schema definition language such that a LOM XML instance will strictly conform	N	tokens, only;¶
979	to IEEE 1484-12.1-2002. The lonStrict.xsd composite XSD		Does not express the uniqueness
000			Deleted: defined by the LOMv1.0 base schema; and¶
980	- Limits vocabularies (source/value pairs) to those defined in <u>Clause 5 of this Stan-</u>		<#>Defines a mechanism to describe
981	<u>dard;</u>	l l	Extensions not the LOMVI.0 base schema. Extensions must be defined in a set of
982	 Expresses the uniqueness constraints defined <u>in Clause 5 of this Standard</u>; and 	1	supporting XSDs. A validating parser will check that the extensions used in a
983	 <u>Does</u> not permit extensions to the LOMv1.0 base schema (see 5.1.3). 		LOM instance conform to rules defined in the supporting XSDs that include ex- tansions
984	<u>C.1.2 IomCustom.xsd</u>		NOTE—The absence of the enforcement of vocabulary values does not relieve a particular LOM instance from satisfying
985 986 987	The lomCustom.xsd composite XSD encodes a particular set of binding <u>constraints in the</u> <u>W3C XML Schema definition language such that a LOM XML instance will <i>conform</i> to IEEE 1484-12 1-2002. The lomCustom xsd composite XSD</u>		vocabulary requirements defined in the LOMv1.0 base schema. Applications that require the use of vocab/loose.xsd component XSD should enforce those
707	TELE 1404 12.1 2002. The Tomed's composite ASD		C.1.1.3 Strict composite XSD ¶
988	- Defines a mechanism to describe vocabularies (source/value pairs) in addition to		The lomStrict.xsd composite XSD
989	those defined in Clause 5 of this Standard. A validating parser will check that vo-		Deleted: the LOMy1 0 base schema
990	<u>cabulary values in a LOM XML instance conform to either the vocabularies de-</u>		only
991	<u>fined in Clause 5 or those defined in an extended vocabulary;</u>		Deleted: by LOMv1.0 base schema
992	 Expresses the uniqueness constraints defined in Clause 5 of this Standard; and 		Deleted: does
993	- Defines a mechanism to describe extensions to the LOMv1.0 base schema (see		Deleted: any
994 995	5.1.3). Extensions are be defined in a set of supporting XSDs. A validating parser will check that the extensions used in a LOM XML instance conform to con-		Deleted: C.1.1.4 Custom composite XSD
996	straints defined in the supporting XSDs that include extensions.		Deleted: rules
997 998	C.1.3 lomLoose.xsd The lomLoose.xsd composite XSD encodes a particular set of binding constraints in the W2C XML Scheme definition language which to LOM XML instances are a side		Deleted: will conform to IEEE 1484-
999	w 3C XML Schema definition language such that a LOM <u>XML</u> instance <u>may not</u> either <u>con</u> -		12.1-2002. The lomCustom.xsd com-
1000	Joint of strictly conjoint to there 1484–12.1–2002. The routhoose .xsd composite ASD	N. N	Defines a mechanism to describe custom
1001	– Does not provide validation to any defined vocabulary values (i.e., values defined	A.	vocabularies (source/value pairs) in addi- tion to the vocabularies defined in the
1002	in Clause 5 of this Standard or values defined in an extended vocabulary). A vali-		LOMv1.0 base schema. A validating
1003	dating parser will verify that the vocabulary values are valid tokens, only;	1.1	in a LOM instance conform to
1004	- Does not express the uniqueness constraints defined in Clause 5 of this Standard;		Deleted: the vocabularies expressed in
1005	and	- X	the LOMv1.0 base schema or those ex-
1006	- Defines a mechanism to describe extensions to the LOMv1.0 base schema (see	Ň	Deleted: Expresses
1007	5.1.3). Extensions are defined in a set of supporting XSDs. A validating parser		Deleted: by the LOMv1 0 base schema:
1008	will check that the extensions used in a LOM XML instance conform to con-	1	and¶
1009	straints defined in the supporting XSDs that include extensions.		sions to the LOMv1.0 base schema. Ex- tensions must be defined in a set of
1010	NOTE—The absence of the validation of vocabulary values does not relieve a particular LOM		supporting XSDs. A validating parser will check that the extensions used in a
1011			sheek that the extensions used in a

Deleted: or part element

Deleted: component XSD defines

Deleted: The

Deleted: schema

- 1012 Applications that require the use of vocab/loose.xsd component XSD should enforce those
- 1013 <u>vocabulary requirements by other means.</u>

1014 C.2 Component XSDs

		Deleted: IEEE 1484.12.1–2002
1015	A component XSD represents a constituent of a composite XSD. The following component $i' \frac{i'}{j'}$	Deleted: The
1016	XSDs are used by the composite XSDs defined in C.1. l_{μ}^{μ}	Deleted: component XSD defines
1017	- common/anyElement xsd: Defines the base XML Schema model groups for $\int_{0}^{1} \frac{d_{1}}{d_{1}} \frac{d_{1}}{d_{$	Deleted: IEEE 1484.12.1–2002.
1018	elements and attributes used for extended data elements and XML attributes.	Deleted: parent
1010		Deleted: element
1019	- common/dataTypes.xsd: <u>Defines</u> global <u>data</u> type declarations for data types $\int_{0}^{1} \int_{0}^{1} \int_{$	Deleted: multiplicity
1020	- common/elementNames.xsd; Defines global element declarations for each of	Deleted: of no more than one item within a LOM instance
1022	the data elements defined in <u>Clause 5 of this Standard</u> . This component XSD is $\frac{1}{1} \frac{1}{10}$	Deleted: The
1023	used to check for the uniqueness of data elements declared to be unique within	Deleted: component XSD defines
1024	their <u>aggregate elements</u> by the presence of the uniqueElementName attribute.	Deleted: schema
1025	The XML Schema constraint unique is used to enforce uniqueness constraints. $\int \frac{d^2}{dt^2}$	Deleted: IEEE 1484.12.1–2002.
1026	- common/elementTypes.xsd: Defines global data type declarations for data	Deleted: relationship
1027	elements defined in Clause 5 of this Standard. This component XSD defines the	Deleted: LOMv1.0 base schema
1028	aggregation relationships among the LOM data elements. These aggregation rela-	Deleted: IEEE 1484.12.1–2002
1029	tionships enforce the LOMv1.0 base schema requirement that elements can be	Deleted: only
1030	present in a LOM XML instance as elements of the aggregate element to which	Deleted: The
1031	they belong <u>, only</u> .	Deleted: component XSD defines
1032	- common/rootElement.xsd: Defines the element name declaration for the ele-	Deleted: in IEEE 1484.12.1–2002
1033	ment that contains all other LOM data elements for a LOM XML instance.	Deleted: The
1034	- common/vocabTypes xsd: Defines global data type declarations for those $\frac{1}{2}$	Deleted: component XSD defines
1035	LOM data elements that have values taken from a vocabulary data type defined	Deleted: IEEE 1484.12.1–2002.
1036	Clause 5 of this Standard.	Deleted: vocabulary value
1037	common (weachWelling, wead). Defines the standard veesbulary value declare	Deleted:
1037 1038 1039	common/vocabvalues.xsq: Defines the standard Vocabulary value declarations are used in con- tions as defined in <u>Clause 5 of this Standard</u> . These declarations are used in con- junction with both vocab/custom.xsd and vocab/strict.xsd.	Deleted: <#>The com- mon/rootElement.xsd component XSD defines the element name declara- tion for the element that contains all other
1040	- <u>extend/custom.xsd</u> : Defines the XML Schema model groups	<pre><#>The common/anyElement.xsd</pre>
1041	customElements and customAttributes to support validation of extended	component XSD defines a global declara- tion that is re-used for extension data
1042	the model groups defined in common (areal) among and This component XSD	elements.¶
1045	should be used if extensions are to be supported in LOM XML instances (see	<#>The unique/loose.xsd compo- nent XSD defines attribute group declara-
1044	5.1.3) Note: Assuming proper processing of the PSVI LOM XML instances that	tions for data elements to support the schema-based validation of uniqueness
1045	use the extend/custom xsd component XSD and extensions will conform but	constraints within a LOM instance where
1047	not strictly conform to this Standard.	the exact set of attributes associa[[29]
1040		Deleted: The
1048	- extend/strict.xsd: Defines the <u>XML Schema</u> model groups	Deleted: component XSD defines
1049	customElements and are used with other component VSDs to support the	Deleted: content
1030	empty model groups and are used with other component ASDS to support the	Deleted: strict

1051 validation of strictly conforming LOM XML instances. This component XSD 1052 should be used if extensions are not to be supported in LOM XML instances. 1053 Note: Assuming proper processing of the PSVI, by using the extend/strict.xad component XSD and, therefore, not supporting extensions. LOM XML instances. Will strictly conform to this Standard. 1056 = unique/loose_xad: Defines XML. Schema model group declarations (in this ease, the XML Schema construct attributes sociated with each element has to be as specified by the LOM XML Schema binding (i.e., when extra attributes to enforce uniqueness have to be avoided). This component XSD is used to relax the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniquees constraints to avoid the introduction of the XML attribute uniquees constraints defined in Clause 5 of this Standard. 1067 force those uniqueness constraints defined in Clause 5 of this Standard. 1068 unique/strict.xad: Defines XML Schema model group declarations (in this tass, the XML. Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard. 1069 unique/strict.xad: Defines XML Schema model group declarations (in this ease, the XML. Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard. 1068 unique/strict.xad: Defines XML Schema model group declarations (in this ease, the XML. Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard and appears with a nulpilicity of at most one (see E.3.3). Note: For most applications, enforcing uniupiue. 107				
1052 should be used if extensions are not to be supported in LOM XML instances. Note: Assuming proper processing of the PSVI. by using the extend/xtrict.xsd component XSD and, therefore, not supporting exten- sions. LOM XML instances will strictly conform to this Standard. 1055 sinns. LOM XML instances will strictly conform to this Standard. 1056 - prigue/Loose.xsd: Defines XML Schema model group declarations (in this support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attributes associated with each element has to be as specified by the LOM XML. Schema binding (i.e., when extra attributes to enforce uniqueness have to be avoided). This component XSD is used to relax the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). Note: The absence of the enforce- ment of uniqueness constraints defined in Clause 5 of this Standard. Applications that use of the unique/Loose.xsd component XSD have to en- force those uniqueness constraints by other means. 1069 - unique/Etrict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construit attribute userGoup) for LOM data element de fined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM XML instance by introducing the attribute uniqueElementName for each LOM XML instance by introducing the attribute uniqueElementName for each LOM Adat element Mas constraints using the uniqueElementName is undesirable, it is unlikely to cause problems. 1071 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended	1051		validation of <i>strictly conforming</i> LOM XML instances. This component XSD	De
1053 Note: Assuming proper processing of the PSVI, by using the extend/strict.xad component XSD and, therefore, not supporting extensions, LOM XML instances will strictly conform to this Standard. 1055 sions, LOM XML instances will strictly conform to this Standard. 1056 unique/locae.xad; Defines XML Schema model group declarations (in this) case, the XML Schema construct attributeGroup) for LOM data elements to support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attributes associated with each element has to be as specified by the LOM XML.Schema binding (i.e., when extra attributes to enforce uniqueness have to be avoided). This component XSD is used to relax the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. 1066 Applications that use of the unique/locae.xad component XSD have to enforce those uniqueness constraints by other means. 1076 case, the XML Schema construct attributeGroup for LOM data elements defined in Clause 5 of this Standard. 1071 unique/strict.xad: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup for LOM data elements defined in Clause 5 of this Standard. 1072 unique/strict.xad: Defines XML Schema model group declarations (in the uniquees constraints within a LOM XML, instance by introducing the attribute unique strict.xad: component XSD is desirable. Al- 1073 to q at most one (see E.3.3). Note: For most applications, enforcing uniqueess	1052		should be used if extensions are not to be supported in LOM XML instances.	De
1054 extend/strict.xsd component XSD and, therefore, not supporting extensions, LOM XML instances will <i>strictly conform</i> to this Standard. 1056 = unique/lose.xsd: Defines XML Schema model group declarations (in this support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attributes associated with each element has to be as specified by the LOM XML Schema binding (i.e., when extra attributes to be asspecified by the LOM XML Schema binding (i.e., when extra attributes to be asspecified by the LOM XML Schema binding (i.e., when extra attributes to be asspecified by the LOM XML Schema binding (i.e., when extra attributes associated with each element has to be asspecified by the LOM XML Schema binding (i.e., when extra attributes associated with each element has the enforcement of uniqueness constraints does not relive a LOM XML instance from satisfying the uniqueness constraints defined in Clause 5 of this Standard. 1066 the enforcements are seed component XSD have to enforce those uniqueness constraints by other means. 1076 case, the XML Schema construct attribute@croup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance from suitable. Allowing the unique/strict.xad component XSD is desirable. All though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 = vocab/custom.xad. Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in a nextended vocabulary. This component XSD is desirable under some component XSD relaxes the vocab/loses.xad component XSD is desirable under some some suitous sing the vocab/custom.xad component XSD is desirable under so	1053		<i>Note:</i> Assuming proper processing of the PSVI by using the	co
 sions_LOM XML instances will strictly conform to this standard. unique/loose.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements to support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attribute@roup) for LOM data element has to be as specified by the LOM XML Schema binding (i.e., when extra attributes on enforce uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). <i>Note:</i> The absence of the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). <i>Note:</i> The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique/strict.xsd: Doff. unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique/strict.xsd: component XSD is desirable. Although adding the attribute uniqueElementName is undesirable, it is unlikely to case groblems. vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. <i>Note</i>: Strict atherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd: component XSD is desirable under some circumstances, but may complicate the schema validation process. vocab/custom .xsd: Enforces that vocabulary values are chara	1054		extend/strict xsd component XSD and therefore not supporting exten-	to
 anique/loose.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements to support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attributes associated with each element has to be as specified by the LOM XML Schema binding (i.e., when extra attributes to enforce uniqueness have to be avoided). This component XSD is used to relax the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). <i>Note:</i> The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique/strict.xsd: Defines XML instance by introducing the attribute uniqueElementName for each LOM XML instance validation of the unique.strict.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data element MSD is desirable. Alto though adding the attribute uniqueElementName is undesirable, it is unlikely to casue problems. vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. <i>Note:</i> Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd: Enforces that vocabulary. <i>Note:</i> Strict adherence to the vocabulary values are character strings. This component XSD relaxes the yalidation process. vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD	1055		sions LOM XML instances will strictly conform to this Standard	scl
1056 - unique/locse.xsd: Defines XML Schema model group declarations (in this support the schema-based validation of uniqueness constraints within a LOM XML instance where the exact set of attributes associated with each element has to be as specified by the LOM XML. Schema binding (i.e., when extra attributes to enforce uniqueness constraints to avoid the introduction of the XML instance where the exact set of attributes associated with each element has to enforce uniqueness constraints does not relieve a LOM XML instance form satisfying the unique element has the set of a tribute associated with each element has the enforcement of uniqueness constraints does not relieve a LOM XML instance form satisfying the unique element has the set of a tribute associated with each element has the unique element has the unique / loose.xsd component XSD have to enforce those uniqueness constraints defined in Clause 5 of this Standard. 1066 - unique / strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique element Name for each LOM XML instance by introducing the attribute uniqueElementName for each LOM adta element that appears with a multiplication group declarations, and the unique element has the standard and unique element tase problems. 1077 - vocab/cuatom.xad: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable. Although adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1078 - vocab/cuatom.xad: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to	1000		sions, Dow Alve historices with strictly conform to this Standard.	lar
1057 case, the XML Schema construct attribute@roup) for LOM data elements to 1058 support the schema-based validation of uniqueness constraints within a LOM 1059 XML instance where the exact set of attributes associated with each element has 1061 to be as specified by the LOM XML Schema binding (i.e., when extra attributes 1061 to enforce uniqueness constraints to avoid the introduction of the XML 1062 the enforcement of uniqueness constraints to avoid the introduction of the XML 1063 attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. 1066 Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. 1067 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attribute@roup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique/strict.xsd component XSD is desirable. Although adding the unique/strict.xsd component XSD is desirable. Although adding the attribute uniqueElementName is undesirable, it is unlikely to casuse problems. 1071 uniqueElementName for each LOM data element to those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable. Although adding the attribute uniqueElementName is undesirable, it is unlikely to casuse problems. 1076 vocab/custom.xsd: Enforces that vocabulary. Note: S	1056	_	unique/loose.xsd: Defines XML Schema model group declarations (in this /	
1058 support the schema-based validation of uniqueness constraints within a LOM Particle 1059 XML instance where the exact set of attributes associated with each element has 1060 to be as specified by the LOM XML Schema binding (i.e., when extra attributes 1061 to enforce uniqueness have to be avoided). This component XSD is used to relax 1062 the enforcement of uniqueness constraints to avoid the introduction of the XML 1063 attribute uniqueelmentName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. 1064 Applications that use of the unique/locse.xsd component XSD have to enforce those uniqueness constraints by other means. 1067 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data element that appears with a multiplicity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Although adding the attribute uniqueElementName for each LOM data element Mate proceed the relation of the support values and relates the vocabulary. Nate: Strind and and extended vocabulary. Nate: Strind add and extended vocabulary walues are only those specified in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable. Although adding the attribute uniqueElementName of standard plus extended vocabulary values using the vocab/custom.xs	1057		case, the XML Schema construct attributeGroup) for LOM data elements to	
1059 XML instance where the exact set of attributes associated with each element has 1060 to be as specified by the LOM XML Schema binding (i.e., when extra attributes 1061 to be as specified by the LOM XML Schema binding (i.e., when extra attributes 1062 the enforce uniqueness have to be avoided). This component XSD is used to relax 1063 attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints does not relieve a LOM XML instance from satis-fying the uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. 1065 fying the uniqueElementName (see E.3.3). Note: The absence of the enforce encore the xat Standard to support schema-based validation of the unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data element that appears with a multiplic-ity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xad component XSD is desirable. Althouge problems. 1077 - vocab/custom.xad: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable. Althouge y values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values schemes and values to be arbitary character strings_LOM XML instances that use the vocab/loose.x	1058		support the schema-based validation of uniqueness constraints within a LOM	De a c
1060 to be as specified by the LOM XML Schema binding (i.e., when extra attributes to enforce uniqueness have to be avoided). This component XSD is used to relax the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints does not relieve a LOM XML instance from satis-fring the unique.tess constraints does not relieve a LOM XML instance from satis-fring the uniqueness constraints does not relieve a LOM XML instance from satis-fring the unique.sc constraints does not relieve a LOM XML instance from satis-fring the unique.sc constraints by other means. 1066 Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. 1067 construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the unique.ElementName for each LOM data element that appears with a multiplic-ty of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Altoward the adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary Nate: Strict adherence to the combined use of standard plus strended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 of from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values are character strings. LOM XML instances that use of the vocab/loose.xsd component XSD may be non-conforming. Applications that require yalidation of the vocabulary value so if the vocab/loose. xsd component X	1059		XML instance where the exact set of attributes associated with each element has	
1061 to enforce uniqueness have to be avoided). This component XSD is used to relax 1062 the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints defined in Clause 5 of this Standard. 1064 Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints defined in Clause 5 of this Standard. 1066 Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. 1067 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplication through adding the attribute unique/strict.xsd component XSD is desirable. Allored though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 of this Standard and extended vocabulary values are character strings. This component XSD is desirable under some circumstances, but may complicates the schema validation process. 1078 vocab/loose.xsd component XSD may be non-conforming. Applic	1060		to be as specified by the LOM XML Schema binding (i.e., when extra attributes	co
1062 the enforcement of uniqueness constraints to avoid the introduction of the XML attribute uniqueElementName (see E.3.3). Note: The absence of the enforcement of uniqueness constraints does not relieve a LOM XML instance from satisfying the uniqueness constraints defined in Clause 5 of this Standard. Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. 1066 unique(strict.xsd): Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM XML instance by introducing the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1074 unique:strict.xsd): Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1085 - yocab/loose.xsd Cmoronent XSD may be non-conforming. Applications for using strings. This component XSD may be non-conforming. Applications for using strings. This component XSD may be non-conforming. Applications for using strings. This component XSD may be non-conforming. Applications for using strings. This component XSD may be non-conforming. Applications for usingsthy values to be arbitrary character strings. L	1061		to enforce uniqueness have to be avoided). This component XSD is used to relax	for
1063 attribute uniqueElementName (see E.3.). Note: The absence of the enforce- ment of uniqueness constraints defined in Clause 5 of this Standard. Applications that use of the unique/loose.xsd component XSD have to en- force those uniqueness constraints dy other means. 1066 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements de- fined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data elements de- tined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the validation process. 1080 vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to proces, the PSVI. 1081 vocab/loose.xsd component XSD andes to palications the vocab/loose.xsd compo	1062		the enforcement of uniqueness constraints to avoid the introduction of the XML	scl
1064 ment of uniqueness constraints does not relieve a LOM XML instance from satis- fying the uniqueness constraints defined in Clause 5 of this Standard. Perform the standard of the unique/loose.xsd component XSD have to en- force those uniqueness constraints by other means. 1067 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements de- fined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance by introducing the attribute unique ElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in uniques 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1081 - yocab/loose.xsd component XSD may be non-conforming. Applications that require yalidation of the souree/value pairs will have to proces, the PSVI	1063		attribute uniqueElementName (see E.3.3). Note: The absence of the enforce-	vo
1065 fying the uniqueness constraints defined in Clause 5 of this Standard. Applications that use of the unique/loose.xsd component XSD have to en- force those uniqueness constraints by other means. 1066 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements de- fined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.3). <i>Note:</i> For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. <i>Note:</i> Strict adherence to the vocabulary values defined in Clause 5 of this Standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. <i>Note:</i> Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1085 vocab/loose.xsd component XSD relaxes the validation process. 1086 vocab/loose.xsd component XSD relaxes the validation process. 1087 sources and values to be arbitrary character strings. LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require_validation of the sourcevalue pairs will have to process, the PS	1064		ment of uniqueness constraints does not relieve a LOM XML instance from satis-	De
1066 Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means. 1067 unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements deminication of the uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplicity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element XSD is desirable. Altonog adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable under some values by checking that both sources and values are taken from either a token set vocabulary values defined in Clause 5 of this Standard and extended vocabulary values by checking that both sources and values are character strings. This component XSD relaxes the validation process. 1081 vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. LOM XML instances that use the vocab/loose.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1082 vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. LOM XML instances that use the vocab/loose.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1083 vocab/loose.xsd: component XSD may be non-conforming.	1065		fying the uniqueness constraints defined in Clause 5 of this Standard.	tio ch
1067 force those uniqueness constraints by other means. 1068 - unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplicity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Although adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable. Although sub checking that both sources and values are taken from either a token set defined in Clause 5 of this Standard and in an extended vocabulary. Note: Strict adherence to the vocabulary values by checking that both sources and values are character strings. This component XSD is desirable under some circumstances, but may complicates the schema validation process. 1080 - vocab/lose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation groess. 1081 defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1082 vocab/lose.xsd component XSD may be non-conforming. Applications that require validation of the vocabulary value validation of the source/value pairs will have to process the	1066		Applications that use of the unique/loose.xsd component XSD have to en-	fro
1068 - unique/strict.xsd: Defines XML Schema model group declarations (in this case, the XML Schema construct attributeGroup) for LOM data elements defined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplicity of at most one (see E.3.3). <i>Note:</i> For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Altot though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD is desirable. Altot fined in Clause 5 of this Standard and in an extended vocabulary. This component XSD referees that vocabulary. Note: Strict adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1085 - yocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation process. When the vocab/loose.xsd component XSD allowing both sources and values to be arbitrary character strings. LOM XML instances that use the vocab/loose.xsd component XSD approxes and values vocabulary value validation of the source/value pairs will have to process the PSVI. 1086 - wocab/loose.xsd component XSD does not relieve a particular LOM XML int	1067		force those uniqueness constraints by other means.	Th
1069 case, the XML Schema construct attributeGroup) for LOM data elements de- fined in Clause 5 of this Standard to support schema-based validation of the uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.). <i>Note:</i> For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. <i>Note:</i> Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1086 - vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI 1090 Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- stance from satisfying the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1068	_	unique/strict.xsd: Defines XML Schema model group declarations (in this $\frac{1}{100}$	scl
1070fined in Clause 5 of this Standard to support schema-based validation of the uniqueElementName for each LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.). <i>Note:</i> For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems.1077-vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. <i>Note:</i> Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1085-yocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings. LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVI Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML instances that use the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- stance from satisfying the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means. <td>1069</td> <td></td> <td>case the XML Schema construct attributeGroup) for LOM data elements de-</td> <td>De</td>	1069		case the XML Schema construct attributeGroup) for LOM data elements de-	De
1071 uniqueness constraints within a LOM XML instance by introducing the attribute uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. Image: Constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. Image: Strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVL Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVL Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD desense not relieve a particular LOM XML in- toppeer vocab/loose.xsd	1070		fined in Clause 5 of this Standard to support schema-based validation of the	wł
1072 uniqueElementName for each LOM data element that appears with a multiplic- ity of at most one (see E.3.3). Note: For most applications, enforcing uniqueeness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. Image: Constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. Image: String	1071		uniqueness constraints within a LOM XML instance by introducing the attribute	De
1073ity of at most one (see E.3.3). Note: For most applications, enforcing uniqueness constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems.1077-vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.Mr1086-yocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1080Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose vocabulary constraints by other means.	1072		uniqueElementName for each LOM data element that appears with a multiplic-	tha
1074constraints using the unique/strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems.1076constraints using the unique.strict.xsd component XSD is desirable. Al- though adding the attribute uniqueElementName is undesirable, it is unlikely to cause problems.1077-vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1084circumstances, but may complicates the validation constraints by allowing both sources and values to be arbitrary character strings_LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVI.1089Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD have to enforce those vocabulary constraints by other means.	1073		ity of at most one (see E 3.3) <i>Note:</i> For most applications enforcing uniqueness	va ne
1075Consume and the attribute unique/former formed to be defined in the problems.1076vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces_adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values by checking that both sources and values are taken from either a token set vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1084vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings_LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVI.1090Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD have to enforce those vocabulary constraints by other means.	1074		constraints using the unique/strict xsd component XSD is desirable Al-	De
1075 Integrated and the difference of the combined use of standard plus extended vocabulary to cause problems. 1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1085 - vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings_LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI. 1090 Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in-top walle vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1075		though adding the attribute unique Flement Name is undesirable, it is unlikely to	(co
1077 - vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process. 1085 - yocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI. 1090 Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- tance from satisfying the vocabulary constraints described in Clause 5 of this Standard. Applications that use the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1076		cause problems	De
1077-vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1085-vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1090Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- stance from satisfying the vocabulary constraints described in Clause 5 of this Standard. Applications that use the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1070			N0 1-
1078Clause 5 of this Standard and in an extended vocabulary. This component XSD enforces adherence to the combined use of standard plus extended vocabulary values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1085-vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process, the PSVI.1090Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1077	_	vocab/custom.xsd: Enforces that vocabulary values are only those specified in $\frac{1}{10}$	un
1079enforces enforces adherence to the combined use of standard plus extended values by checking that both sources and values are taken from either a token set defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1085-vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings_LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1089Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- tocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1078		<u>Clause 5 of this Standard and in an extended vocabulary. This component XSD</u> $\frac{1}{10} \frac{1}{10} \frac{1}$	pa the
1080values by checking that both sources and values are taken from either a token set1081defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the1082vocabulary values defined in Clause 5 of this Standard and extended vocabulary1083values using the vocab/custom.xsd component XSD is desirable under some1084circumstances, but may complicates the schema validation process.1085vocab/loose.xsd: Enforces that vocabulary sources and values are character1086strings. This component XSD relaxes the validation constraints by allowing both1087sources and values to be arbitrary character strings. LOM XML instances that use1088the vocab/loose.xsd component XSD may be non-conforming. Applications1089that require validation of the source/value pairs will have to process the PSVI.1090Note: The absence of the enforcement of the vocabulary value validation by the1091vocab/loose.xsd component XSD does not relieve a particular LOM XML in-1092stance from satisfying the vocabulary constraints described in Clause 5 of this1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1079		enforces adherence to the combined use of standard plus extended vocabulary in the intervention of standard	the
1081defined in Clause 5 or from an extended vocabulary. Note: Strict adherence to the vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1084vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1089Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose.xsd component XSD does not relieve a particular LOM XML in- toose vocabulary constraints by other means.	1080		values by checking that both sources and values are taken from either a token set $\frac{1}{1} \frac{1}{10} \frac{1}{10}$	De
1082vocabulary values defined in Clause 5 of this Standard and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1084vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1089Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- stance from satisfying the vocabulary constraints described in Clause 5 of this Standard. Applications that use the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1081		defined in <u>Clause 5 or from an extended vocabulary. Note: Strict adherence to the</u> $\int_{u_{ij}}^{u_{ij}}$	De
1083values using the vocab/custom.xsd component XSD is desirable under some circumstances, but may complicates the schema validation process.1084vocab/loose.xsd: Enforces that vocabulary sources and values are character strings. This component XSD relaxes the validation constraints by allowing both sources and values to be arbitrary character strings, LOM XML instances that use the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1089Note: The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- stance from satisfying the vocabulary constraints described in Clause 5 of this Standard. Applications that use the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.	1082		vocabulary values defined in Clause 5 of this Standard and extended vocabulary	/ ur
1084 circumstances, but may complicates the schema validation process. 1085 - vocab/loose.xsd: Enforces that vocabulary sources and values are character 1086 strings. This component XSD relaxes the validation constraints by allowing both 1087 sources and values to be arbitrary character strings. LOM XML instances that use 1088 the vocab/loose.xsd component XSD may be non-conforming. Applications 1089 that require validation of the source/value pairs will have to process the PSVI. 1090 Note: The absence of the enforcement of the vocabulary value validation by the 1091 vocab/loose.xsd component XSD does not relieve a particular LOM XML in- 1092 stance from satisfying the vocabulary constraints described in Clause 5 of this 1093 Standard. Applications that use the vocab/loose.xsd component XSD have to 1094 enforce those vocabulary constraints by other means.	1083		values using the vocab/custom.xsd component XSD is desirable under some	De
 1085 - <u>vocab/loose.xsd</u>: Enforces that vocabulary sources and values are character 1086 strings. This component XSD relaxes the validation constraints by allowing both 1087 sources and values to be arbitrary character strings. LOM XML instances that use 1088 the vocab/loose.xsd component XSD may be non-conforming. Applications 1089 that require validation of the source/value pairs will have to process the PSVI. 1090 Note: The absence of the enforcement of the vocabulary value validation by the 1091 vocab/loose.xsd component XSD does not relieve a particular LOM XML in- 1092 stance from satisfying the vocabulary constraints described in Clause 5 of this 1093 Standard. Applications that use the vocab/loose.xsd component XSD have to 1094 enforce those vocabulary constraints by other means. 	1084		circumstances, but may complicates the schema validation process. $\frac{1}{2}$	str 2–
1086strings. This component XSD relaxes the validation constraints by allowing both1087sources and values to be arbitrary character strings, LOM XML instances that use1088the vocab/loose.xsd component XSD may be non-conforming. Applications1089that require validation of the source/value pairs will have to process the PSVI.1090Note: The absence of the enforcement of the vocabulary value validation by the1091vocab/loose.xsd component XSD does not relieve a particular LOM XML in-1092stance from satisfying the vocabulary constraints described in Clause 5 of this1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1085	_	vocab/loose.xsd: Enforces that vocabulary sources and values are character $\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	un
1087sources and values to be arbitrary character strings, LOM XML instances that useDefendence1088the vocab/loose.xsd component XSD may be non-conforming. ApplicationsDefendence1089that require validation of the source/value pairs will have to process the PSVI.Defendence1090Note:The absence of the enforcement of the vocabulary value validation by theDefendence1091vocab/loose.xsd component XSD does not relieve a particular LOM XML in-Defendence1092stance from satisfying the vocabulary constraints described in Clause 5 of thisDefendence1093Standard. Applications that use the vocab/loose.xsd component XSD have toDefendence1094enforce those vocabulary constraints by other means.Defendence	1086		strings. This component XSD relaxes the validation constraints by allowing both	
1088the vocab/loose.xsd component XSD may be non-conforming. Applications that require validation of the source/value pairs will have to process the PSVI.1089Note:1090Note:1091vocab/loose.xsd component XSD does not relieve a particular LOM XML in-1092stance from satisfying the vocabulary constraints described in Clause 5 of this1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1087		sources and values to be arbitrary <u>character</u> strings, <u>LOM XML instances that use</u>	6-
1089that require validation of the source/value pairs will have to process the PSVI.1090Note:1090Note:1091Vocab/loose.xsd component XSD does not relieve a particular LOM XML in-1092stance from satisfying the vocabulary constraints described in Clause 5 of this1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1088		the vocab/loose.xsd component XSD may be non-conforming. Applications	De
1090Note:The absence of the enforcement of the vocabulary value validation by the vocab/loose.xsd component XSD does not relieve a particular LOM XML in- tance from satisfying the vocabulary constraints described in Clause 5 of this Standard. Applications that use the vocab/loose.xsd component XSD have to enforce those vocabulary constraints by other means.Description Description	1089		that require validation of the source/value pairs will have to process the PSVI.	vo
1091vocab/loose.xsd component XSD does not relieve a particular LOM XML in-1092stance from satisfying the vocabulary constraints described in Clause 5 of this1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1090		Note: The absence of the enforcement of the vocabulary value validation by the	De
1092stance from satisfying the vocabulary constraints described in <u>Clause 5 of this</u> 1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1091		vocab/loose.xsd component XSD does not relieve a particular LOM XML in-	De
1093Standard. Applications that use the vocab/loose.xsd component XSD have to1094enforce those vocabulary constraints by other means.	1092		stance from satisfying the vocabulary constraints described in Clause 5 of this	
1094 enforce those vocabulary constraints by other means.	1093		Standard. Applications that use the vocab/loose.xsd component XSD have to	
	1094		enforce those vocabulary constraints by other means.	

Deleted: standard data elements.

Deleted: The vocab/custom.xsd component schema enforces strict conformance with the standard vocabulary values specified in the LOMv1.0 base schema and any custom-defined vocabulary extensions. The vocab/custom.xsd enforces strict

Deleted: custom

Deleted: IEEE 1484.12.1–2002 or from custom token set.

Deleted: The vocab/strict.xsd component schema enforces strict conformance with the standard vocabulary values specified in the LOMv1.0 base ichema, but it does not allow custom vocabulary values from other sources.

Deleted: schema supports strict validation of standard vocabulary values by checking that both sources and values are from a token set defined in IEEE 1484.12.1–2002. ¶ The vocab/loose.xsd component schema

Deleted: enforcement of the vocabulary source/value relationship constraints, which simplifies the schema

Deleted: process for those applications that perform vocabulary source/value validation by other means. This component schema relaxes the strict validation

Deleted: of the vocab/strict.xsd component schema

Deleted: ¶ NOTES: ¶

1—The absence of the enforcement of uniqueness constraints does not relieve a particular LOM instance from satisfying the uniqueness constraints described in the LOMv1.0 base schema.

Deleted: that

Deleted: the use of the

unique/loose.xsd component XSD

Deleted: enforce those uniqueness constraints by other means.¶ 2—For most applications, enforcing the uniqueness constraints using the unique/strict.xsd compo[...[30]]

Deleted: .¶

6-Strict adherence to

Deleted: standard vocabularies using the vocab/strict.xsd compon ... [31]

Deleted: source/

Deleted: relationship constraints

Deleted: schema

Deleted: the LOMv1.0 base schema.

Deleted: schema

39

ments of an aggregate data element, only, and do not define data types or value spaces for aggregate data elements.¶ These requirements imply that any aggregate data element should be allowed to contain additional custom data elements. This requirement can be met by including an appropriate <xs: any> element in the content model for aggregate elements (<xs:any namespace= "##other" proc-

The processContents = "lax" declaration instructs an XML processor to attempt to validate the element's content. This declaration allows elements from namespaces other than the LOM namespace to be included and their contents to be validated, if schema information for the elements can be found. If such information is not available, the XML processor will allow any wellformed LOM instance.¶

1—To ensure that valid LOM instances are created, an organization that develops extensions should also provide an XSD that defines the validation rules for the

2—Organizations providing extensions are free to define namespaces for these extensions. If an organization provides a namespace, then the XSDs provided by this standard may have to be altered to support the extended element namespace.¶

essContents="lax"/>).¶

NOTES:¶

extensions.¶

1095	- vocab/strict.xsd: Enforces that vocabulary values are only those specified in	Deleted: C.3 Enabling extensions¶
1096	Clause 5 of this Standard. This component XSD does not allow extended vocabu-	With certain constraints, the LOMv1.0 base schema anticipates the need to intro-
1097	lary values from other sources. This component XSD supports the validation of	duce data elements not included in the
1098	strictly conforming vocabulary values by checking that both sources and values	standard collection of elements. These constraints are¶
1099	are from a token set defined in Clause 5.	<#>Extensions retain the original value spaces and data types of LOMv1.0 base schema data elements; and¶ <#>Extensions provide additional ele-

1100 **Annex D**

1101	(informative)
1102	Enabling extended data elements and attributes
1102	

1103	The LOMv1.0 base schema anticipates the need to introduce extensions not included in the
1104	standard collection of elements (see 5.1.3). These extensions have to meet the following con-
1105	straints:
1106	 Extensions to the LOMVv1.0 base schema retain the value spaces and data types
1107	of LOM data elements from the LOMv1.0 base schema; and
1108	- Extensions do not define data types or value spaces for aggregate elements in the
1100	LOMv1 0 base schema
1107	Lonry, ouse seneme.
1110	These requirements imply that if an aggregate element contains extensions, the extensions do
1111	not redefine the intended meaning (i.e., value space and data type) of the aggregate element.
1112	To meet these requirements, the informative schemas (see Annex B) include an any element
1113	in the content model for aggregate elements (<xs:any <="" namespace="##other" td=""></xs:any>
1114	processContents="lax"/>).
1115	The processContents="lax" declaration instructs an XML processor to attempt to vali-
1116	date the element's content. This declaration allows elements from namespaces other than the
1117	LOMv1.0 base schema namespace to be included and their contents to be validated, if schema
1118	information for the elements can be found. If such information is not available, the XML
1119	Schema processor will validate any well-formed LOM XML instance.
1120	NOTES:
1121	<u>1—To ensure that conforming LOM XML instances are created, an organization that develops</u>
1122	extensions also provide an XSD that defines the validation constraints for the extensions.
1123	2-Organizations providing extensions may choose to define namespaces for these extensions. If
1124	an organization defines a namespace, then the XSDs provided by this Standard may have to be
1125	altered to support the namespace.

1126 D.1 Enabling extended data elements

1127	LOMv1.0 base schema data elements may be extended. These extensions are new XML ele-
1128	ments defined in a namespace other than the namespaces defined in 5.2. Extended data ele-
1129	ments may be added to any LOM data element that is defined as an aggregate element.
1130	To enable extended data elements, the extend/custom.xsd component XSD should be

1131 used in the construction of the composite XSD. This component XSD defines the XML 1132 Schema model group customElements that references the customElements model group

1133 1134 1135	defined in the common/anyElement.xsd component XSD. (See the lomCustom.xsd file provided with this Standard, [Annex B].) The inclusion of these two component XSDs by a composite XSD enables the validation of extended data elements.
1136	The relevant component XSDs and how they enable extended data elements are described be-
1137	low.
1138	 <u>common/anyElement.xsd</u>: Defines the base XML Schema model group for
1139	elements used for extended data elements.
1140	 <u>common/dataTypes.xsd and common/elementTypes.xsd</u>: For each aggre-
1141	gate complexType declaration defined in this component XSD, an XML Schema
1142	model group reference to the customElements declaration is defined.
1143	 <u>common/vocabTypes.xsd</u>: For each vocabulary complexType declaration de-
1144	fined in this component XSD, an XML Schema model group reference to the
1145	customElements declaration is defined.
1146	 <u>extend/custom.xsd</u>: Defines the XML Schema model group
1147	customElements to support validation of extended data elements. The XML
1148	Schema model group in this component XSD references the model group defined
1149	in the common/anyElement.xsd component XSD.
1150 1151	NOTE—Extensions may be disabled by using the extend/strict.xsd composite XSD.
1152 1153 1154	LOMv1.0 base schema data elements may be extended with XML attributes defined in a namespace other than the namespaces defined in 5.2. Extended XML attributes may be added to any LOM data element.
1155	To enable extended attributes, the extend/custom.xsd component XSD should be used in
1156	the construction of the composite XSD. This component XSD defines the XML Schema
1157	model group customAttributes. This model group defines the ability to add any attribute
1158	from a namespace other than the namespaces defined in 5.2. The inclusion of this component
1159	XSD by a composite XSD enables the validation of extended attributes.
1160	The relevant component XSDs and how they enable extended attributes are described below.
1161	 <u>common/anyElement.xsd</u>: Defines the base XML Schema model group for
1162	elements used for extended data elements.
1163	 <u>common/dataTypes.xsd</u> and <u>common/elementTypes.xsd</u>; For each aggre-
1164	gate <u>complexType</u> declaration defined in this component XSD, an XML Schema
1165	model group reference to the <u>customAttributes</u> declaration is defined.
1166	 <u>common/vocabTypes.xsd</u>: For each vocabulary and value complexType dec-
1167	laration defined in this component XSD, an XML Schema attributeGroup

1168 1169		reference to the customAttri defined.	butes XM	IL Schema	a model gro	oup declar	<u>ation is</u>
1170 1171	_	<pre>extend/custom.xsd: Defi customAttributes to suppor</pre>	nes the that ion	XML of extend	Schema ed attributes	<u>model</u> 3.	group
1172	<u>NOTE-</u>	-Extensions may be disabled by us	ing the exte	end/stric	t.xsd com	posite XSD	<u>.</u>

1173 **Annex E**

1174 (informative)

1175 XSD implementation choices

1176Major implementation choices contained in the example XSDs (see Annexes B and C) are
discussed in $\underline{E}.1 - \underline{E}.6.$

1178 **E.1 Data types**

11/0		51		
1179 1180 1181 1182 1183 1184 1185	The common/dataTypes.xsd component XSD defines a collection of global W3C XML Schema definition language type declarations for the data types used to constrain values for the data elements defined in <u>Clause 5 of this Standard</u> . The type declarations are provided so that the logical data types in the LOMv1.0 base schema are defined in terms of their underlying XML Schema data type once, only. This modularization of the schema definition provides a means to change the underlying schema data type, if necessary, without making any other changes to the schema definition.			
1186	The following global W3C XML Schema definition language types are defined:	1		
1187	 CharacterString: an alias for the XML Schema primitive data type string. 			
1188	 LanguageId: an alias for the XML Schema derived data type language 			
1189	 VCard: an alias for CharacterString. 	'¦\ ''(
1190	 MimeType: an alias for CharacterString. 	/(
1191	 Size: an alias for the XML Schema derived data type nonNegativeInteger. 	',(
1192 1193 1194	 LanguageString: a sequence of zero or more string elements with an option- ally defined language attribute. The language attribute describes the language of the value held by the string element. 			
1195	 DateTime: optional subelements dateTime and description. 	1		
1196	- Duration: optional subelements duration and description	/		
1197 1198	LanguageId is used to constrain the values of the various language elements. Note that the General category's language elements use the related type LanguageIdOrNone (see 5.4.1).	 / /		
1199 1200 1201	The LanguageString type allows each string element to contain the optional attribute language of type LanguageId, which is defined as the XML Schema derived data type language.			
1202 1203	The DateTime type has the optional subelements dateTime of type CharacterString and description of type LanguageString. The CharacterString type for the			

Deleted: D
Deleted: D
Deleted: D
Deleted: IEEE 1484.12.1–2002.
Deleted: only
Deleted: xs:
Deleted: xs:
Deleted: ¶ <#>LanguageIdNone: an alias for xs:token.¶ LanguageIdOrNone: a union of the LanguageID and LanguageId- None.
Deleted: xs:
Deleted: LangString
Deleted: <
Deleted: >
Deleted: VCard and MimeType are data types that act as placeholders. These data types may be further defined in future editions of this standard.¶
Deleted: LangString
Deleted: <
Deleted: >
Deleted: <
Deleted: >
Deleted: <
Deleted: >
Deleted: LangString.

1204 1205	dateTime subelement contains a restricted pattern of characters. The pattern, defined using a regular expression in W3C XML Schema definition language is	Deleted: <
1206 1207 1208 1209	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	Deleted: restricted
1210	The restricted character string pattern is used instead of the XML Schema primitive data type	Deleted: xs:dateTime
1211	because of the restrictions placed on the value space defined in IEEE 1484,12.1-2002. These	Deleted: 11
1212	restrictions cannot be enforced by the XML Schema primitive data type dateTime.	Deleted: This restricted pattern
1010		Deleted: xs:dateTime
1213	The Duration data type has the optional subelements duration of type	Deleted: <
1214	CharacterString, and description, of type LanguageString. The	Deleted: >
1215	acters. The pattern, defined using a regular expression in W3C XMI. Schema definition lan-	Deleted:
1210	guage, is	Deleted: <
1218	$= \frac{1}{2}$	Deleted: >
1210	$9]{1,}H){0,1}([0-9]{1,}M){0,1}([0-9]{1,}()){0,1}{0,1}{0,1}{0,1}{0,1}{0,1}{0,1}{0,1}$	Deleted: the same
1220	9]{1,}){0,1}S){0,1}){0,1}	Deleted: as the <description> element elsewhere.</description>
1221	The restricted character string pattern is used instead of the XML Schema primitive data type	Deleted:
1222	because of the restrictions placed on the value <u>space</u> defined in IEEE 1484, <u>12</u> .1–2002., <u>These</u>	Deleted: <
1223	restrictions cannot be enforced by the XML Schema primitive data type duration XML	Deleted: >
1224	Schema data type.	Deleted: xs:duration
		Deleted: 11
1225	E.2 Elements	Deleted: This restricted pattern
		Deleted: xs:duration
1226 1227 1228 1229 1230 1231	The common/elementNames.xsd component XSD defines a collection of global W3C XML Schema definition language <u>data</u> type declarations for the LOM data elements. The component XSD contains 78 type declarations, one for each LOM data element. Duplicate type declarations for elements that share element names are included within XML comments for completeness. Those elements included within XML comments should be left in XML comments.	Deleted: D
1232 1233 1234 1235	The global schema type declarations are provided so that the composite schema is defined in terms of a collection of underlying types, rather than being defined directly in terms of XML Schema constructs. This modularization of the schema definition provides a means to change the definition of the elements without making any other changes to the schema definition.	
1236	For each I OM data element an XML element name is assigned by the convention of using	Deleted: spelling
1230	camel-case capitalization of the full name of the LOM data element with an initial lower-case	Deleted: provided with
1238	letter (lowerCamelCase) (see 5.4 and 1.1).	Deleted: a global W3C XML Schema definition language
1239	Each LOM data element is given a global W3C XML Schema definition language element	Deleted: a global W3C XML Schema definition language
1240	declaration, type declaration, and attribute group declaration. The XML names defined in	Deleted: Schema element name is
1241	<u>Clause 5 are</u> used for the <u>names</u> of each of these declarations.	Deleted: name

Deleted: D

Typically, the W3C XML Schema definition language type declaration for an element will provide the content model for the element, referring to the global element declarations for subelements as needed and including the attributes from the W3C XML Schema definition language attribute group. LOMv1.0 base schema aggregate elements will have declarations that are represented by complex types. LOMv1.0 base schema simple elements will have declarations that are represented by simple types.

1248	E.3 Aggregates		
1249 1250 1251	The selection of the W3C XML Schema definition language structure for aggregate elements involves satisfying three competing requirements that arise either directly or indirectly from the LOMv1.0 base schema. These are		Deleted: data
1252 1253	a) An ordering restriction is not defined in IEEE 1484.12.1–2002. Each element should allow arbitrary ordering of its subelements.		
1254 1255	b) Each element should enforce multiplicity constraints defined in IEEE 1484.12.1–2002 on its subelements.		
1256 1257 1258	c) The aggregation structure should be preserved without the introduction of <u>addi-</u> <u>tional</u> container elements (i.e., elements that do not map directly to those defined in IEEE 1484.12.1–2002, but are introduced to meet the first two requirements).		Deleted: artificial
1259 1260 1261 1262 1263	Requirements (a) and (b) <u>above</u> introduce complexity in expressing the content model for ag- gregate elements. Requirement (c) <u>above</u> arises from the implied constraint that the content model should match the LOMv1.0 base schema as closely as possible. The W3C XML Schema definition language does not provide a way to satisfy requirements (a) and (b) without the introduction of <u>additional</u> container elements (c).		Deleted: artificial
1264 1265	Three alternatives for the selection of the content model for aggregate elements could be implemented.		
1266 1267	<u>NOTE—In E.3.1 – E.3.5, "a", "b", and "c" in parentheses or square brackets refer to the require</u> ments listed above.		Deleted: D
		1	Deleted: <xs:< td=""></xs:<>
1268	E.3.1 Using the XML Schema element sequence	1.1	Deleted: >
12(0			Deleted: <xs:< td=""></xs:<>
1269	The use of the XML <u>Schema element</u> sequence <u>provides</u> a way to meet the requirements of multiplicity (b) and avoiding additional container elements (c) but does not provide arbitrary	<u> </u>	Deleted: > construct defines
1270	ordering (a). If subelements are required to appear in a specific order, the XSD can easily en-		Deleted: artificial containers
1272	force the multiplicity constraints without introducing <u>additional</u> container elements. Figure <u>E.1</u>		Deleted: artificial
1273	shows how the sequence <u>element</u> can be used to define the LOMv1.0 base schema.		Deleted: D
1274			Deleted: XML
1275		Ň	Deleted: construct <xs:sequence></xs:sequence>
1276	<xs:complextype name="lom"></xs:complextype>		
1277 1278	<xs:sequence> <xs:element minoccurs="0" ref="general"></xs:element></xs:sequence>		

1279	<xs:element minoccurs="0" ref="lifeCycle"></xs:element>		
1280	<xs:element minoccurs="0" ref="metaMetadata"></xs:element>		
1281	<xs:element minoccurs="0" ref="technical"></xs:element>		
1282	<xs:element <="" maxoccurs="unbounded" minoccurs="0" td=""><td></td><td></td></xs:element>		
1283	rel="educational"/>		
1285	<pre><xs:element maxoccurs="unbounded" minoccurs="0" ref="relation"></xs:element></pre>		
1286	<xs:element <="" maxoccurs="unbounded" minoccurs="0" td=""><td></td><td></td></xs:element>		
1287	ref="annotation"/>		
1288	<xs:element <="" maxoccurs="unbounded" minoccurs="0" td=""><td></td><td></td></xs:element>		
1289	ref="classification"/>		
1290			
1292	,	1	Deleted: D
1293	Figure F 1—An example using sequence construct	11	Deleted: <xs:< td=""></xs:<>
1275		2	Deleted: >
1294	According to Figure <u>E.</u> 1 the general element can appear zero or one time (satisfies [b]). The		
1295	example does not introduce additional container elements (satisfies [c]). However, the gen-		Deleted: artificial
1296	eral element, if present, has to precede the lifeCycle element (does not satisfy [a]).		Deleted: must
1207	LOW YNU instance spectral units this supervise the same have		Deleted: lifecycle
1297	LOM XML instances created using this approach may be non-conforming.		Deleted: D
1298	E.3.2 Using the XML Schema element all		Deleted: <xs:all> ¶ The use of</xs:all>
			Deleted: <xs:< td=""></xs:<>
1299	The XML Schema element all provides a way to meet the requirements of arbitrary ordering		Deleted: > construct defines
1300	(a). Using the all element requires the introduction of additional container elements (c) to		Deleted:) and multiplicity (b). However
1301	meet the requirement of multiplicity (b). The all element requires the introduction of addi-	11	Deleted: <xs:< td=""></xs:<>
1302	tional container elements to collect multiple instances of a given element. The LOMv1.0 base		Deleted: > construct introduces
1303	schema contains elements that can have multiplicities of more than one (e.g., Educational).	Sec. 1	Deleted: use
1304	Figure <u>E</u> .2 illustrates how the all <u>element</u> can be used to define the LOMv1.0 base schema.		Deleted: artificial
1305			Deleted: XML <xs:< td=""></xs:<>
1206		$\frac{1}{1}$	Deleted: > construct
1306	<rs:complextype name="lom"></rs:complextype>		Deleted: artificial
1308	<xs:all></xs:all>	~ 10	Deleted: D
1309	<xs:element minoccurs="0" ref="general"></xs:element>	- N	Deleted: XML sequence construct
1310	<pre><xs:element minoccurs="0" ref="lifeCycle"></xs:element> <us:element minoccurs="0" ref="meteMetedate"></us:element></pre>	Ì	<xs:< td=""></xs:<>
1312	<pre><xs:element minoccurs="0" ref="metaMetadata"></xs:element> <xs:element minoccurs="0" ref="technical"></xs:element></pre>		Deleted: >
1313	<pre><xs:element minoccurs="0" ref="educationals"></xs:element> <!-- container--></pre>		
1314	<xs:element minoccurs="0" ref="rights"></xs:element>		
1315	<xs:element minoccurs="0" ref="relations"></xs:element> container		
1310	<pre><xs:element minoccurs="0" ref="annotations"></xs:element> <!-- container--></pre>		
1317	<pre><xs:element minoccurs="0" rel="classifications"></xs:element> </pre>		
1319			
1320			Deleted D
1321		1	
1322	Figure E.2—An example using the all element	12	Deleted: >
			Deleted: >
I			

05 November 2004

1323	According to Figure <u>E.</u> 2 the general element can appear zero or one time (<u>satisfies [b]</u>) any-		
1324	where in the sequence of elements (<u>satisfies [a]</u>). However, the example introduces <u>additional</u>		Deleted: artificial
1325	container elements, such as the educationals element (does not satisfy [c]), where the plu-		Deleted: (
1326	ral educationals indicates zero or more educational elements.		
1227	LOM YML instances created using this enpressed may be new conforming	1	Deleted: D
1327	LOW ANIL instances created using this approach may be non-conforming.		
		1	Deleted: <xs:< td=""></xs:<>
1328	E.3.3 Using the XML Schema element choice	1-1	Deleted: >
			Deleted: construct defines
1329	The use of the XML Schema element choice provides a way to meet the requirements of		
1330	arbitrary ordering (a) and avoiding <u>additional</u> containers (c) but does <u>not</u> enforce multiplicity		Deleted: artificial
1331	constraints (b), By allowing any subelement to appear any number of times, the W3C XML		Deleted: A more flexible solution can
1332	Schema definition language types can easily be constructed to allow arbitrary ordering of the		be found by temporarily giving up on the
1333	subelements without having to introduce <u>additional</u> container elements. Figure <u>E</u> .3 shows how		straints.
1334	the choice element can be used to define the LOMv1.0 base schema.	S	Deleted: artificial
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		Deleted: D
1335		1	Deleted: XML sequence construct
1336			<xs:< td=""></xs:<>
1337	<xs:complextype name="lom"></xs:complextype>		Deleted: >
1338	<xs:choice maxoccurs="unbounded" minoccurs="0"></xs:choice>		
1339	<xs:element ref="general"></xs:element>		
1340	<xs:element ref="lifeCycle"></xs:element>		
1341	<xs:element ref="metaMetadata"></xs:element>		
1342	<xs:element ref="technical"></xs:element>		
1343	<rs:element ref="educational"></rs:element>		
1344	<xs:element ref="rights"></xs:element>		
1345	<xs:element ref="relation"></xs:element>		
1346	<xs:element ref="annotation"></xs:element>		
1347	<rs:element ref="classification"></rs:element>		
1348			
1349			Deleted: D
1550			
1351	Figure E.3—An example using <u>the choice element</u>	<u>11 -</u>	
			Deleted. >
			Deleted: artificial
1352	The example in Figure <u>E.3</u> does not introduce <u>additional</u> container elements (satisfies [c]) and		()
1353	allows arbitrary ordering (satisfies [a]). However, all of the elements may appear any number		
1354	of times (does not satisfy [b]).		
1355	This solution accepts a superset of what the LOMv1.0 base schema allows without introduc-		
1356	ing additional complexity in the LOM XML instances or the LOM XSD. For applications		
1357	willing to enforce I OM data element multiplicity constraints on the PSVI this may represent		Deleted: uniqueness
1359	a sufficient solution	<0	Deleted of the White Mar
1330	a summer solution.		Deleted: post-schema validation XML infoset
1359	LOM XML instances created using this approach may be non-conforming.		Deleted: D

1360	E.3.4 Using the XML Schema element choice with the XML		Deleted: <xs:< th=""></xs:<>
1361	Schema constraint unique		Deleted: >
1001	n		Deleted: <xs:< th=""></xs:<>
1362	The use of the XML Schema element choice with the XML Schema constraint unique		Deleted: >
1363 1364 1365	provides a way to meet the requirements of arbitrary ordering (a) and avoiding additional con- tainers (c) and enforces multiplicity constraints (b). Using this approach, the multiplicity con- straints can be enforced using the <u>XML Schema</u> constraint unique. For each element with a		<b>Deleted:</b> The example in Figure D.3 does not enforce multiplicity constraints (b) for those elements that can appear zero or one time only. The
1366	multiplicity of at most one in its <u>aggregate element</u> , define <u>an</u> attribute called		Deleted: identity
1367	uniqueElementName, with a fixed default value equal to the name of the element, and use		Deleted: <xs:< td=""></xs:<>
1368	this attribute as the element key for unique constraint. Figure E.4 shows how the unique		Deleted: >
1369	<u>constraint</u> can be used to enforce multiplicity constraints for element uniqueness when the		Deleted: parent
13/0	LOMVI.0 base schema dennes that an element cannot appear more than one time.		Deleted: a hidden
1371			Deleted: "
1272		$\frac{\mu_{11}}{\mu_{11}}$	Deleted: "
1372	<xs:element name="lom" type="lom"></xs:element>		Deleted: <xs:< td=""></xs:<>
1374	<xs:unique name="lomUnique"></xs:unique>	$\frac{1}{1}$	Deleted: >.
1375	<pre><xs:selector xpath="*"></xs:selector> <xs:field xpath="@uniqueFlementName"></xs:field></pre>		Deleted: D
1377 1378	<pre> </pre>		Deleted: XML sequence construct
1379			Deleted: >
1380	Figure E.4—An example using the unique constraint		Deleted: D
1000			Deleted: <xs:< td=""></xs:<>
			Deleted: >
1381	One such constraint using the unique constraint is needed for each aggregate element that		Deleted: construct <xs:< td=""></xs:<>
1382	contains subelements with multiplicities of at most one.		Deleted: >
1383	The disadvantages of this approach are the introduction of the unique Element Name attrib-	Ì	Deleted: parent
1384	ute and the introduction of additional complexity in the XSD.		Deleted: "
	·	<u>`</u>	Deleted: "
1385	This solution maximizes the amount of validation that can be done by a parser. For applica-	Ì	Deleted: a limited amount of
1386	tions that <u>cannot process the PSVI</u> , the addition of the uniqueElementName attribute may		<b>Deleted:</b> are not influenced by
1387	<u>be</u> a sufficient solution.		Deleted: "
1388	This solution supports the validation of both <i>conforming</i> and strictly conforming LOM XMI		Deleted: "
1389	instances.		Deleted: during parsing, this
		Ň	Deleted: represent
1390	F.3.5 Summary	È.	Deleted: D
1201			
1391	Subclauses E.3.1 – E.3.4 defined alternative approaches to aggregate elements. A summary of		Deleted: D
1392	those alternatives is given below		Deleted: D
1373			Deleted: content models.
1394	- sequence: Does not satisfy requirement (a), which states that the subelements		Deleted: <xs:< td=""></xs:<>
1395	appear in an arourary order. This alternative does not conform to IEEE 1484.12.1- 2002		Deleted: >

1397	- all: Does not satisfy requirement (c), which preserves the aggregation structure		Deleted: <xs:< td=""></xs:<>
1398	defined in IEEE 1484.12.14-2002. The use of the XML Schema element all re-		Deleted: >
1399	quires the introduction of arbitrary container elements (e.g.,		Deleted: <xs:< td=""></xs:<>
1400	educationals). This alternative requires all subelements of the container to be		Deleted: >
1401	confected under the <u>agglegate</u> element.		Deleted: <
1402	- choice: Does not enforce uniqueness constraints. All of the elements may appear		Deleted: >
1403	any number of times. This alternative does not satisfy requirement (b) and does	$\langle \rangle$	Deleted: parent container
1404	not conform to IEEE 1484.12.1-2002. This alternative is defined in the $\sqrt{2}$	$\langle \cdot \rangle$	Deleted: <xs:< td=""></xs:<>
1403	unique/100se.xsd component ASD (see C.2).	``	Deleted: >
1406	- choice, with unique: Overcomes the problems with the XML Schema element		Deleted: <xs:< td=""></xs:<>
1407	choice. An attribute is applied to each element with a multiplicity of at most		Deleted: >
1408	one. The attribute is enforced by applying an identity constraint. This alternative		Deleted: <xs:< td=""></xs:<>
1409	is defined in the unique/strict.xsd component ASD (see C.2).	Ň	Deleted: > attribute
		Ň	Deleted: <xs:< td=""></xs:<>
1410	<u>E</u> .4 Vocabularies	Ń	Deleted: >. A hidden
		, i	Deleted: hidden
1411	The selection of the content model for <u>LOM</u> data elements <u>of type Vocabulary</u> involves satis-		Deleted: D
1412	fying three requirements that arise either directly or indirectly from the LOMv1.0 base schema		Deleted: vocabulary
1413	and from now vocabularies are to be used by applications. These are		
1414	a) Data elements of type Vocabulary should allow arbitrary values (source/value		Deleted: Vocabulary
1415	pairs).		Deleted: from arbitrary sources
1416	b) If the source of a vocabulary value is the LOMv1.0 base schema (i.e.,		Deleted: 0
1417	<pre><source/>LOMv1.0</pre> , then the token used to represent the vocabulary		Deleted: <\
1418	value is defined in Clause 5 of this Standard.		Deleted: space
1419	c) If the source of a vocabulary value is not the LOMv1 0 base schema (i.e., not		<b>Deleted:</b> described by the LOMv1.0 base schema
1420	<pre><source/>LOMv1.0</pre> , then the token used to represent the vocabulary		Deleted: 0
1421	value should not conflict with any of the tokens defined in Clause 5 of this Stan-		Deleted: <\
1422	<u>daru</u> .		Deleted: space
1423	The W3C XML Schema definition language cannot easily accommodate the goal of strict	,	Deleted: the LOMv1.0 base schema
1424	type checking for vocabulary values in a one-pass model. It can provide validation of a fixed		Deleted: Unfortunately, the
1425	set of vocabularies, <u>but cannot validate arbitrary vocabularies</u> . In LOM <u>XML</u> instances, the		<b>Deleted:</b> that will meet the needs of the wide range of applications entipipated for
1426	lixed vocabularies are defined based on the source of the vocabulary. If the source of the vo- ashulary is defined $W_{2C}$ XML Scheme definition language constructs can be defined to aid		the LOM XML Schema binding.¶
1427	in the validation of the vocabularies	N _N	The W3C XML Schema definition lan- guage
1120		$\langle \rangle$	Deleted: only.
1429	Three alternatives for expressing LOM vocabularies in W3C XML Schema definition lan-	$\langle \rangle$	Deleted: being made available
1430	guage are presented in $E.4.1 - E.4.3$ . These three alternatives provide options on the validation		Deleted:
1431	approaches taken by XML Schema processors.	( (	Deleted: D
1432	NOTE—In E.4.1 – E.4.3, "a", "b", and "c" in parentheses refer to the requirements listed above $\sqrt{2}$		Deleted: D
		$\langle \rangle$	Deleted: how

Copyright © 2004 IEEE. All rights reserved.

Deleted: are Deleted: D

### 1433 **<u>E</u>.4.1 vocab/loose.xsd**

1434 If <u>no requirements exist</u> for validation of vocabulary values from the LOMv1.0 base schema 1435 vocabularies, then the vocab/loose.xsd component <u>XSD</u> may be used. <u>This</u> component 1436 <u>XSD</u> provides a way to meet the goal of allowing values from arbitrary sources (a) but does 1437 not validate that the vocabulary value is a valid member of the set of values defined by the 1438 <u>source (b and c)</u>. This approach is the simplest alternative in terms of the complexity of the 1439 XSD. Each element that takes its values from a vocabulary is defined as a 1440 CharacterString as shown in Figure <u>E.5</u>.

- 1441
- 1442
- 1443 1444 1445

1446 1447

#### Figure E.5—An example from the vocab/loose.xsd component XSD

Both the source and the value <u>elements</u> are optional and may appear in any order. There 2 are no constraints on vocabulary data elements.

1450This approach accepts a superset of what the LOMv1.0 base schema allows without introduc-1451ing additional complexity in LOM XML instances. The disadvantage of this approach is that it1452does not support validation of vocabulary values. Using the vocab/loose.xsd component1453XSD requires processing the PSVI to validate vocabulary values used in both strictly conform-1454ing and conforming LOM XML instances.

### 1455 **E.4.2 vocab/strict.xsd**

1456 If vocabularies are limited to only those defined in the LOMv1.0 base schema, then the 1457 vocab/strict.xsd component XSD may be used. This component XSD defines con-1458 straints for a validation of LOMv1.0 base schema vocabularies. The use of this component 1459 XSD requires the source to be LOMV1.0 (i.e., <source>LOMV1.0</source>) and the value to be a valid LOM-defined vocabulary (b) but does not allow values from arbitrary sources (a 1460 1461 and c). This approach is a straightforward implementation of the values allowed by the vo-1462 cabulary elements. As shown in Figure E.6, each element that takes its values from a vocabu-1463 lary is defined where the source element is given a fixed value of LOMV1.0, and the value of 1464 the value element is derived from an enumerated list of the appropriate vocabulary values. 1/65

1405	
1466	<xs:simpletype name="difficulty"></xs:simpletype>
1467	<pre><xs:union membertypes="lom:difficultyValues"></xs:union></pre>
1468	
1469	
1470	<xs:complextype name="difficultyVocab"></xs:complextype>
1471	<xs:choice maxoccurs="unbounded" minoccurs="0"></xs:choice>
1472	<xs:element name="source" type="sourceValue"></xs:element>

Deleted: requirement exists
Deleted: schema
Deleted: The . vocab/loose.xsd
Deleted: schema
Deleted: provide validation.
Deleted: provide validation. Deleted:

Deleted: D
Deleted: using
Deleted: <
Deleted: >
Deleted: <
Deleted: >
Deleted: the
Deleted: or the LOMv1.0 base schema
Deleted: allow
Deleted: schema
Deleted: other validation steps
Deleted: the
Deleted: D
Deleted: with
<b>Deleted:</b> a strict vocabulary validation approach
<b>Deleted:</b> desirable. The vo- cab/strict.xsd
Deleted: schema
Deleted: rules
Deleted: strict
Deleted: the vocab/strict.xsd
Deleted: schema
Deleted: <\
Deleted: standard
Deleted: D
Deleted: <
Deleted: >
Deleted: <
Deleted: >

1473	<xs:element name="value" type="difficultyValue"></xs:element>		
14/4	<xs:group ref="ex:customElements"></xs:group>		
1475			
14/6	<pre><xs:attributegroup ref="ex:customAttributes"></xs:attributegroup></pre>		
14//			
14/8			
14/9	<pre><xs:simpletype name="difficultyValues"></xs:simpletype></pre>		
1460	<pre><xs:restriction pase="xs:token"></xs:restriction></pre>		
1401	<pre><xs:enumeration value="very easy"></xs:enumeration> </pre>		
1482	<pre></pre>		
1484	<pre><xs:enumeration value="mediant"></xs:enumeration></pre>		
1485	<pre><rs:enumeration <rs:enumeration="" value="very difficult"></rs:enumeration></pre>		
1486			
1487			Deleted: D
1488	() HD DIMPICIATE	1	
1.00		11	Deleted: using
1489	Figure <u>E</u> .6—An example <u>from the</u> vocab/strict.xsd <u>component XSD</u>	ji ,	Deleted: <
		1	Deleted: >
		11	
1490	Both the source and value elements are optional and may appear in any order. Values are	11-1-	Deleted: <
1/01	validated against the vocabulary tokens defined in Clause 5 of this Standard for the LOMy1.0	2	Deleted: >
1402	has scheme		Deleted:
1492	base schema.	125	Deleted: standard
1493	This approach accepts a subset of what the LOMv1.0 base schema would allow by excluding		Deleted: standard
1494	vocabulary values with sources other than the LOMv1.0 base schema. The advantage of this		Deleted: the XSD
1495	approach is that it can validate the vocabulary values defined in this Standard.	j.	Deleted: D
		1	Deleted VC
1.10.6		11	Deleted: If
1496	E.4.3 Vocab/custom.xsd	(-)	Deleted: limited to both those
1.407		111	Deleted: with
1497	To use both vocabularies that are defined in the LOMv1.0 base schema and extended vocabu-	12	Deleted: a custom defined set of
1498	laries, the vocab/custom.xsd component <u>XSD may be used</u> . This component <u>XSD</u> defines		Deleted: n a custom vocabulary valida-
1499	<u>constraints</u> for validation of both LOMv1.0 base schema vocabularies and vocabularies that	1	tion approach may be desirable. The
1500	are defined in other schemas. The use of this component XSD provides a way to meet the goal		Deleted: schema
1501	of allowing values from arbitrary sources (a) by supporting the validation of both vocabulary	11	Deleted: rules
1502	values from other sources (c) and LOMv1.0 base schema vocabulary values (b). This ap-		Deleted: by
1503	proach provides additional flexibility in the partial validation of both standard and extended	(1, 1)	Deleted: the weach (quation wad
1504	vocabulary values. As shown in Figure E.7 each element that takes its values from a vocabu-	(1,1)	Deleted. Inc vocab/ custom.xsu
1505	lary is defined where the source element is allowed to have an arbitrary value, and the value	(1,1,1)	Deleted: schema
1506	of the value element is derived from a custom list of the appropriate vocabulary values.		<b>Deleted:</b> but does not meet the requirements of arbitrary ordering (b
1507	<pre>cvc:cimplo™mo_nomo-#difficulty#&gt;</pre>		<b>Deleted:</b> avoiding artificial containers
1509	<pre><xs:union membertypes="lom:difficultvValues lx:difficultvValues"></xs:union></pre>	我的	Deleted: supplies
1510		111	Deleted: custom
1511		$-\frac{1}{2}$	
1512	<pre><xs:complextype name="difficulty"></xs:complextype></pre>	$-\frac{1}{1}$	Deleted: D
1515	<pre><xs.complexcontent></xs.complexcontent></pre>	$-\frac{1}{n}$	Deleted: <
1514	<pre><xs:attributegroup ref="ag:difficulty"></xs:attributegroup></pre>	$\frac{1}{1}$	Deleted: >
1516	<pre> <pre> <pre> <pre>       </pre></pre></pre></pre>	1	Deleted: <
1517			
			Deleteu. >

1518			
1519	<pre>cyclications name="difficultyWelling"&gt;</pre>		
1520	<pre><xs:restriction base="xs:token"></xs:restriction></pre>		
1522	<pre><xs:enumeration value="very easy"></xs:enumeration></pre>		
1523	<xs:enumeration value="easy"></xs:enumeration>		
1524	<xs:enumeration value="medium"></xs:enumeration>		
1525	<xs:enumeration value="difficult"></xs:enumeration>		
1526	<pre><xs:enumeration value="very difficult"></xs:enumeration> </pre>		
1527			
1520		1	Deleted: D
1530	Figure F 7—An example from the vocab/custom xsd component XSD	11	Deleted: using
1550	rigure En An example from the vocab/custom.xsu component xob		
1.50.1			Deleted: <
1531	Both the source and value elements are optional and may appear in any order. Values are	<u> </u>	Deleted: >
1332	validated against the <u>extended</u> vocabiliary enumerations.		Deleted: <
1533	This approach accepts a superset of what the LOMv1.0 base schema allows and allows the use	$\sim$	Deleted: >
1534	of both standard and extended vocabulary values. The advantage of this approach is that it re-	```	Deleted: custom
1535	tains the ability for the validation of standard vocabulary values while accommodating the <u>use</u>		Deleted: custom the
1530	use to those defined in the LOMy10 hase schema when the source is LOMy10 (i.e.	17. 1.	Deleted: presence of custom
1538	$\alpha$ as to $\alpha$ solution in $\alpha$		Deleted: However,
1539	the LOMv10 base schema along with extended vocabulary values if the source is not		Deleted: does not restrict
1540	LOMV1.0.		Deleted: vocabulary values
			Deleted: <\
1541	NOTE—If an organization plans to extend the vocabulary values and is following an XSD design similar to the one defined in this Standard, the organization should define the remarkage that de		Deleted: >). Nor does
1543	fines the extended vocabularies. The vocab/custom yed component XSD provides a place-		Deleted: prevent
1544	holder for this extension (xmlns:lx="http://ltsc.ieee.org/XSD/LOM/custom"). The		Deleted: standard
1545	organization may change this declaration.		Deleted: may
			Deleted: schema
		1	Deleted: D
1546	E.5 Additional notes	1	
1547	This subclause discusses additional decisions that were made to enforce conformance to the	11	Deleted: Some
1548	LOMv1.0 base schema. The following information applies to implementation choices that		Deleted: additional notes apply
1549	were used in this Standard and in the XSD files provided with this Standard (see Annex B).		Deleted: :
1550	These implementation choices are included in the dataTypes.xsd component XSD.		
1551	- Certain elements are defined in IEEE 1484.12.1–2002 to be represented as valid		
1552	vCard syntax. At the time of publication of this Standard, no <i>de facto</i> standard		Deleted: standard, there was
1553	W3C XML Schema definition language binding for the vCard specification was	-	
1554	available. XSD implementations may want to define a type to encapsulate ele-		<b>Deleted:</b> and define a
1555	ments of type VCard. The VCard data type does not validate that the associated	 	<b>Deleted:</b> for those elements that have a
1556	value represents a well-formed Internet Mail Consortium 3.0 vCard (see IETF		value of a vCard.
1557	RFC 2426:1998).		Deleted: vCard
		<b>`</b> ,	Deleted: vCard

53

1558 1559 1560 1561	_	The MimeType global W3C XML Schema definition language type is defined to encapsulate and define a type for those elements that are defined to hold a MIME type. The MimeType type does not validate that the MimeType value represents a valid MIME type (see IETF RFC 2425 [A2]).		
1562 1563 1564 1565 1566	_	A global element type of LanguageIdOrNone has been created to encapsulate the declaration for the Language element (see the General element, 5.4.1). This type allows the Language element to have two values: a Language identifier (as defined in ISO 639 and ISO 3166) or the token none, which indicates no lan- guage.		Deleted: (indicating Deleted: )
1567 1568 1569 1570	_	An element type of LanguageId has been created to encapsulate the declaration for the Language element (see the Educational element, 5.4.5). This type allows the Language element to have a Language identifier (as defined in ISO 639 and ISO 3166) and multiplicity greater than one.		
1571 1572 1573 1574 1575	_	The global <description> element has content model LangString and multi- plicity one. Local element declarations for the Description element (see the General element, 5.4.1) and for the Description element (see the Educational element, 5.4.5) allow these elements to have multiplicity greater than one.</description>		
1576 1577 1578 1579 1580	_	Because the Description element appears with different multiplicities in the LOMv1.0 base schema, two XML elements were created. The global element description has content model LanguageString and multiplicity one. The global element descriptionUnbounded has content model LanguageString and unbounded multiplicity.		Deleted: The global <entity> element Deleted: VCard</entity>
1581 1582 1583 1584 1585	_	Because the Name element is both a subelement of the Taxon Path element (see 5.4.9.2) and the Vocabulary data type (see 5.5.5) and has different data types depending on the Source element's aggregate element, two source declarations were developed. One allows validation of Taxon Path elements, and the other allows validation of vocabulary values.		Deleted: The enumerated values for Deleted: are listed as dependent on the value Deleted: corresponding Type Deleted: the OrComposite element,
			1 1	Field Code Changed

54

**Deleted:** ). The values of the <name> element in the XSD are given as the union of the listed values.¶

A local element declaration for the Source element (see the Taxon path ele-

**Deleted:** ) allows the validation of vocabulary elements to provide a global element declaration for the <source> element that is appropriate for the source

of a vocabulary value.

ment,

Page 9: [1] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Format		
Size		
Location		
Requirement		
Installation		
Remarks		
Other Platform Requirements		
Duration		
Page 9: [2] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Interactivity Type		
Learning Resource Type		
Interactivity Level		
Semantic Density		
Intended End User Role		
Context		
Typical Age Range		
Difficulty		
Typical Learning Time		
Description		
Language		
Page 9: [3] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Cost		
Copyright and Other Restrictions		
Description		
Page 9: [4] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Kind		
Resource		
Page 11: [5] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
	Data	
Page 11: [5] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
	*	
Page 11: [6] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [6] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		

Page 11: [7] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [7] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [7] Deleted	Scott Lewis	11/5/2004 2·09·00 PM
Catalog		
Entry		
Entry		
Page 11: [8] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [8] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [8] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
String		
Page 11: [9] Deleted	Scott Lowic	11/5/2004 2:00:00 PM
	Scott Lewis	117572004 2:09:00 PW
Page 11: [9] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [9] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Unspecified		
Page 11: [10] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [10] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [10] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
String		
Page 11: [11] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Dama 11, [11] Dalakad	Caratte Lauria	11 /F /2004 2:00:00 DM
Page 11: [11] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
~		
Page 11: [11] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
String		
-		
Page 11: [12] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [12] Deleted	Scott Lewis	11/5/2004 2:09:00 PM

~
~

Page 11: [12] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
String		
Page 11: [13] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [13] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [13] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Source		
Value		
Page 11: [14] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [14] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [14] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Source		
Value		
Page 11: [15] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
	Data	
Page 11: [15] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
	*	
Page 11: [16] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
Page 11: [16] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
>		
Page 11: [17] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
<		
<		
< Page 11: [17] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >>	Scott Lewis	11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >>	Scott Lewis	11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >> Page 11: [18] Deleted	Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >> Page 11: [18] Deleted <	Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >> Page 11: [18] Deleted <	Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM
<pre>&lt; Page 11: [17] Deleted &gt;&gt; Page 11: [18] Deleted &lt; Page 11: [18] Deleted</pre>	Scott Lewis Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM
<pre>     Page 11: [17] Deleted     &gt;&gt;     Page 11: [18] Deleted     &lt;      Page 11: [18] Deleted     &gt; </pre>	Scott Lewis Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM
< Page 11: [17] Deleted >> Page 11: [18] Deleted < Page 11: [18] Deleted >	Scott Lewis Scott Lewis Scott Lewis	11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM 11/5/2004 2:09:00 PM

*Common data types and data elements are defined in .

Page 13: [20] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Source		
Value		
Page 13: [21] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
DateTime		
Description		
	<b>•</b>	
Page 13: [22] Deleted	Scott Lewis	11/5/2004 2:29:00 PM
*Common data types and data	elements are defined in .	
Page 15: [23] Deleted	Scott Lewis	11/5/2004 2:41:00 PM
*Common data types and data	alamants are defined in	
Common data types and data	elements are defined in .	
Page 15: [24] Deleted	Scott Lewis	11/5/2004 2:53:00 PM
*Common data types and data	elements are defined in	
Page 15: [25] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
Source		
Value		
Page 15: [26] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
DateTime		
Description		
Page 15: [27] Deleted	Scott Lewis	11/5/2004 2:53:00 PM
*Common data types and data	elements are defined in .	
Page 27: [28] Deleted	Scott Lewis	11/5/2004 3:13:00 PM

e data type is represented as an XML element. The DateTime data type may contain the following subelements:

Date Time Description

If present, the subelements listed above shall be contained by the element for which the DateTime type is defined. The order of appearance of subelements shall not be significant.

### **Date Time**

Table 21 describes the Date Time element and its first-generation subelements.

LOM name XML name		Subelements	Min	Max	Order	Data type
Date	<date></date>	DateTime	0	1	Unspecified	DateTime
		Description				
DateTime	<datetime></datetime>	None	0	1	Unspecified	CharacterString
Description	<description></description>	String	0	1	Unspecified	LangString

Table 21—The DateTime element

#### DateTime

The value space for the Date Time element is a pattern which shall be defined according to the following rules.

Page 38: [29] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
The common/rootE	llement.xsd component XSD	) defines the element name
declaration for the ele	ement that contains all other eler	nents for a LOM instance.
The common/anyEl	.ement.xsd component XSD	defines a global declaration
that is re-used for ext	ension data elements.	
The unique/loose	e.xsd component XSD defines	attribute group declarations
for data elements to	support the schema-based va	lidation of uniqueness con-
straints within a LO	M instance where the exact set	of attributes associated with
each element has to	be precisely as specified by the	ne LOM XML binding (i.e.,
where extra, artificial	l attributes must be avoided). Th	is component XSD is used to
relax the enforcement	it of the uniqueness constraints	to avoid the introduction of
the uniqueElement	Name artificial attribute (see D.3	3.3).
The unique/stric	t.xsd component XSD defines	s attribute group declarations
for data elements def	ined in IEEE 1484.12.1–2002 to	o support schema-based vali-
dation of the unique	less constraints with a LOM inst	tance by introducing the arti-
ficial attribute uniqu	ueElementName for each data	element that appears with a
multiplicity of at mos	st one (see D.3.3).	

The extend/custom.xsd component XSD defines the content model group customElements to support validation of custom data elements.

Page 39: [30] Deleted	Scott Lewis	11/5/2004 2:09:00 PM
-		

enforce those uniqueness constraints by other means.

2—For most applications, enforcing the uniqueness constraints using the unique/strict.xsd component XSD is desirable. Although adding such an artificial attribute is undesirable, it is unlikely to cause problems.

3—LOM instances that use the extend/custom.xsd component XSD and extensions will be *conforming* but not *strictly conforming* to this standard.

4—By using the extend/strict.xsd component XSD and, therefore, not supporting extensions, *conforming* LOM instances will *strictly conform* to this standard.

5—Strict adherence to standard plus custom vocabularies using the vocab/custom.xsd component schema is desirable under some circumstances, but may complicates the schema validation

Page 39: [31] Deleted	Scott Lewis	11/5/2004 2:09:00 PM

standard vocabularies using the vocab/strict.xsd component schema is desirable under some circumstances, but disallowing custom vocabularies is undesirable in other circumstances.

7—