# IEEE P1484.12.3/D5 Draft Standard for Learning TechnologyExtensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata 

Sponsor<br>Learning Technology Standards Committee<br>of the<br>IEEE Computer Society


#### Abstract

This Standard 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 Standard for Learning Object Metadata. The purpose of this Standard is to allow the creation of LOM instances in XML. This allows for interoperability and the exchange of LOM XML instances between various systems. This Standard uses the W3C XML Schema definition language to define the syntax and semantics of the XML encodings.

Keywords: learning object metadata, LOM, LOM XML instance, LOM XML Schema binding, 1484.12.1-2002, Extensible Markup Language, XML, XML Schema definition, XSD, W3C XML Schema definition language, metadata.


The Institute of Electrical and Electronics Engineers, Inc.
Three Park Avenue, New York, NY 10016-5997, USA
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.

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-

Deleted: standard

Deleted: standard

Deleted: standard

Deleted: The
dardization activities, permission must first be obtained from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department. Other entities seeking permission to reproduce this document, in whole or in part, must obtain permission from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department.

IEEE Standards Department
Standards Licensing and Contracts
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
[Note: Information about IEEE LTSC P1484.12.3 can be found at:
http://ltsc.ieee.org/wg12/
This note will be removed upon reaching the final draft of this IEEE document.]

## Introduction

(This introduction is not part of P1484.12.3, Draft Standard for Learning Technology_ Extensible Markup Language (XML) Schema Binding for Learning Object Metadata.)

This Standard 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. This Standard defines the structure and constraints of the XML 1.1 documents in W3C XML Schema definition language.

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 for interoperability and the exchange of LOM XML instances between various systems.

## Participants

At the time this Standard was completed, the working group had the following membership:

| Wayne Hodgins, Chair <br> Erik Duval and Scott Lewis, Technical Editors |  |  |
| :--- | :--- | :--- |
| Brandon Muramatsu | Daniel Rehak | Schawn Thropp |
| Additional names to be sup- <br> plied |  |  |

Deleted: standard
Deleted: a
Deleted: XML Schema definition language binding for

Deleted: data model

Deleted: standard

## Deleted:

The following persons were on the balloting committee:
To be provided by IEEE editor at time of publication,

Deleted: (
Deleted: )

Also included are the following nonvoting IEEE-SA Standards Board liaisons:
To be supplied

## Contents

1. Overview ................................................................................................................................... 1
1.1 Scope ................................................................................................................................... 1
1.2 Purpose ................................................................................................................................ 1
2. Normative references ................................................................................................................. 1
3. Definitions and acronyms ......................................................................................................... 2
3.1 Definitions........................................................................................................................... 2
3.2 Acronyms and abbreviations.............................................................................................. 4
4. Conformance ............................................................................................................................. 4
4.1 Strictly conforming LOM XML instances ......................................................................... 5
4.2 Conforming LOM XML instances .................................................................................... 5
5. LOM XML Schema binding definition ................................................................................... 5
5.1 General information............................................................................................................ 5
5.1.1 Smallest permitted maximums ................................................................................... 6
5.1.2 LOM XML instance conformance constraints ........................................................... 6
5.1.3 Extension support........................................................................................................ 6
5.2 LOM namespaces ............................................................................................................... 7
5.3 Table format and organization ............................................................................................ 8
5.4 LOM .................................................................................................................................... 9
5.4.1 General....................................................................................................................... 10
5.4.2 Life Cycle .................................................................................................................. 12
5.4.3 Meta-Metadata.......................................................................................................... 14
5.4.4 Technical.................................................................................................................. 16
5.4.5 Educational ............................................................................................................... 19
5.4.6 Rights ......................................................................................................................... 22
5.4.7 Relation..................................................................................................................... 23
5.4.8 Annotation ................................................................................................................. 25
5.4.9 Classification ............................................................................................................. 25
5.5 Common data types and elements ................................................................................... 27
5.5.1 CharacterString.......................................................................................................... 27
5.5.2 DateTime ................................................................................................................... 27
5.5.3 Duration .................................................................................................................... 29
5.5.4 LangString .................................................................................................................. 30
5.5.5 Vocabulary................................................................................................................ 32

Annex A...................................................................................................................................... 33
Annex B ..................................................................................................................................... 34
Annex C........................................................................................................................................ 35
C. 1 Composite XSDs.............................................................................................................. 36
C.1.1 lomStrict.xsd ............................................................................................................. 37
C.1.2 lomCustom.xsd......................................................................................................... 37
C.1.3 lomLoose.xsd........................................................................................................... 37
C. 2 Component XSDs ............................................................................................................ 38

Annex D..................................................................................................................................... 41
D. 1 Enabling extended data elements ..... 41
D. 2 Enabling extended attributes ..... 42
Annex E ..... 44
E. 1 Data types ..... 44
E. 2 Elements ..... 45
E. 3 Aggregates ..... 46
E.3.1 Using the XML Schema element sequence. ..... 46
E.3.2 Using. ..... 47
the XML Schema element all ..... 47
E.3.3 Using the XML Schema element choice ..... 48
E.3.4 Using the XML Schema element choice with the XML Schema constraint unique49
E.3.5 Summary ..... 49
E. 4 Vocabularies ..... 50
E.4.1 vocab/loose.xsd ..... 51
E.4.2 vocab/strict.xsd. ..... 51
E.4.3 vocab/custom.xsd ..... 52
E. 5 Additional notes ..... 53

# Draft Standard for Learning Technology- 

 Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata
## 1. Overview

The scope and purpose of this Standard are discussed in 1.1 and 1.2.

### 1.1 Scope

This Standard 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, $\overline{\text { Standard for Learning Object Metadata. }{ }^{1} \text { This Standard defines the structure and constraints }}$ of the XML 1.1 documents in W3C XML Schema definition language. An implementation that conforms to this Standard shall conform to IEEE 1484.12.1-2002.

### 1.2 Purpose

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 for interoperability and the exchange of LOM XML instances between various systems.


## 2. Normative references

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

IEEE 1484.12.1-2002, Standard for Learning Object Metadata.

[^0]IETF RFC 2048:1996, Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures.

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

ISO/IEC 10646-1, Information technology-Universal multiple-octet coded character set Part 1: Architecture and basic multilingual plane.
,W3C Recommendation(2 May 2001), XML Schema Part 1: Structures.
,W3C Recommendation(2 May 2001), XML Schema Part 2: Data types.
W3C Recommendation (4 February 2004), Namespaces in XML 1.1.

## 3. Definitions and acronyms

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

### 3.1 Definitions

For purposes of this Standard, the following terms and definitions apply. IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition $[\mathrm{B} 1]^{1}$, should be referenced for terms not defined in this subclause.
aggregate element: A LOM data element that contains other LOM data elements called subelements. See also: subelement.
component Extensible Markup Language Schema definition (component XSD): An Extensible Markup Language Schema definition that defines a constituent of a composite schema. See also: composite Extensible Markup Language Schema definition.
composite Extensible Markup Language Schema definition (composite XSD): An Extensible Markup Language Schema definition that is a structure made up of distinct component XML Schema definitions. See also: component Extensible Markup Language Schema definition.
content model: A framework that identifies the makeup (i.e., data types, multiplicity constraints, ordering) of a specific model.

[^1]Deleted: IETF RFC 2426:1998, vCard MIME Directory Profile. $\mathbb{T}$

Deleted: standard
Deleted: A1
Deleted: Clause

Deleted: XML
Deleted: represents
Deleted: or part element
Deleted: XML
Deleted: XML
Deleted: XML
data type: A property of distinct values, indicating common features of those values and operations on those values.
extended data element: A data element that is not defined in the LOMv1.0 base schema.
extended vocabulary: A value space that is not defined in the LOMv1.0 base schema for a LOM data element of type Vocabulary.

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.

Extensible Markup Language Schema binding (XML Schema binding): A textual representation of the behaviors, attributes, and value space of a data-model element in W3C XML Schema definition language,
learning object: In this Standard, any entity, digital or non-digital, that may be used for learning, 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 1484.12.12002. See also: LOMv1.0 base schema.
learning object metadata Extensible Markup Language instance (LOM XML 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 Schema binding defined in this Standard.

LOMv1.0 base schema: A structured collection of standard data items, including their data types, multiplicities, and container/component relationships, defined in IEEE 1484.12.1-2002, Clause 6.
mixed content: An element type has mixed content when elements of that type may contain character data, optionally interspersed with child elements (W3C, Extensible Markup Language [XML] 1.1 [B4]).

Multipurpose Internet Mail Extensions type (MIME type): A standard way of classifying content types on the Internet.
post-schema-validation infoset: A transformed version of the XML infoset of a document produced by a conformance W3C XML Schema processor. W3C XML Schema defines the data elements added to the XML infoset of the original document during validation to produce the post-schema-validation infoset. See also: Extensible Markup Language information set.

Deleted: infoset
Deleted: infoset
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 descriptions. See also: data type. ${ }^{\|}$

Deleted: standard

Deleted: In this standard, a
Deleted: standard. See also: LOMv1.0 base schema

Deleted: described
subelement: A LOM data element that is contained within another LOM data element called an aggregate element. A subelement may contain other subelements, in which case it is also an aggregate element. See also: aggregate element.
token: Tokens are character strings. The value space of a token is the set of strings that do not contain the line feed ( $\# \mathrm{xA}$ ) nor tab ( $\# \mathrm{x} 9$ ) characters, that have no leading or trailing spaces ( $\# x 20$ ), and that have no internal sequences of two or more spaces. The lexical space of a token is the set of strings that do not contain the line feed ( $\# \mathrm{xA}$ ) nor tab ( $\# \mathrm{x} 9$ ) characters, that have no leading or trailing spaces ( $\# x 20$ ), and that have no internal sequences of two or more spaces. The base type of a token is normalizedString. For purposes of this Standard, tokens are case sensitive. (Adapted from W3C, XML Schema Part 2.) See also: token set.
token set: A set of tokens in which each token is unique. See also: token.
uniqueness constraint: A restriction placed on a LOM data element that enforces that the 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 object metadata data element.
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 reference to another standard or specification.

### 3.2 Acronyms and abbreviations

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

## 4. Conformance

Conformance to this Standard is discussed in 4.1 and 4.2. In 4.1 and 4.2, strictly conforming LOM XML instance and conforming LOM XML instance refer to the metadata represented in the LOM XML instance prior to any processing of the LOM XML instance.

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

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

Deleted: standard

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. ${ }^{\top}$
<\#>Strictly conforming LOM instances $\|$
A strictly conforming LOM instance $\boldsymbol{\top}$ <\#>Shall conform to the LOMv1.0 base schema requirements of IEEE 1484.12.12002;
Shall conform to the requirements of Clause

## Deleted: ;

Deleted: $\mathbb{\|}$
<\#>Shall not contain any extensions to the LOMv1.0 base schema defined in IEEE 1484.12.1-2002. ${ }^{\text {a }}$
<\#>Conforming LOM instances $\pi$ A conforming LOM instance $\mathbb{\square}$ <\#>Shall conform to the LOMv1.0 base schema requirements of IEEE 1484.12.12002; and
Shall conform to the requirements of Clause

### 4.1 Strictly conforming LOM XML instances

A strictly conforming LOM XML instance

- Shall be a strictly conforming LOM instance as defined in IEEE 1484.12.1-2002;
- Shall conform to the requirements of Clause 5 of this Standard;
- Shall not include vocabulary values that are not defined in Clause 5 of this Standard;
- Shall not include XML elements or attributes that are not defined in Clause 5 of this Standard; and
- Shall not include mixed content.


### 4.2 Conforming LOM XML instances

A conforming LOM XML instance

- 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 by using the extension mechanism described in 5.1.3; and
- May include mixed content.


## 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.

### 5.1 General information

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

The LOMv1.0 base schema defines an aggregation relationship between data items. An XSD for LOM defines an aggregation relationship between subelements that maintains the relationship defined in the LOMv1.0 base schema. In a LOM XML instance, a subelement shall appear only within the aggregate element of the aggregation relationship. For example, in 5.4.3.1, the Identifier subelement appears by definition as a component of the Meta-Metadata

## Deleted: of

Deleted: W3C
Deleted: definition language element for XML

Deleted: All LOMv1.0 base schema

Deleted: parent
element described in 5.4.3. The presence of the subelement automatically implies the presence of the aggregate element to which the subelement belongs.

The LOMv1.0 base schema does not define any sequencing of LOM data elements, except for their aggregation relationships. Therefore, an XML Schema binding of the LOMv1.0 base schema shall not define any sequence of the elements in a LOM XML instance, except for their aggregation relationships.

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

### 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., max0ccurs for aggregate 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 instances against IEEE 1484.12.1-2002 should not use maxOccurs or maxLength restrictions or should set the restricted limit to unbounded.

### 5.1.2 LOM XML instance conformance constraints

Constraints on the validity of a conforming or strictly conforming LOM XML instance with respect to IEEE 1484.12.1-2002 are defined throughout 5.4 and 5.5. All conforming 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.

### 5.1.3 Extension support

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

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 maxOc curs 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. $\|$
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 conforming but not for strictly conforming 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).
- 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).
- 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.


### 5.2 LOM namespaces

All of the LOM data elements defined in 5.4 and 5.5 shall be defined in the namespace http://ltsc.ieee.org/xsd/LOM. This namespace is reserved by this Standard and shall not be used to represent any other data element (see W3C Recommendation, Namespaces in XML 1.1.). This namespace shall not be used to define extensions to a LOM XML instance.

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).

- http://ltsc.ieee.org/xsd/LOM/custom
- http://ltsc.ieee.org/xsd/LOM/unique
- http://ltsc.ieee.org/xsd/LOM/vocab
- http://ltsc.ieee.org/xsd/LOM/extend

If an organization provides extensions to a LOM XML instance, the organization should define the namespace for these extensions.

| Deleted: The following namespaces are <br> reserved by this standard |
| :--- |
| Deleted: used in defining the elements |
| Deleted: this standard:ब |
| The |
| Deleted: name for this standard shall be: |
| Deleted: standard, |
| Deleted: ; ; |
| Deleted: The namespace name prefix <br> for the <br> ht tp: //ltsc . ieee. org/xsd/LOM <br> namespace shall be lom. This value is <br> reserved by this standard, <br> Deleted: the prefix <br> Deleted: namespace; <br> Deleted: The namespace <br> Deleted: shall be reserved for the LOM <br> XML custom composite XSDs (see An- <br> nex B) provided by IEEE; <br> Deleted: The namespace <br> Deleted: shall be reserved for the LOM <br> XML unique composite XSDs (see An- <br> nex B) provided by IEEE; <br> Deleted: The namespace <br> Deleted: shall be reserved for the LOM <br> XML vocab composite XSDs (see Annex <br> B), provided by IEEE; and <br> Deleted: The namespace <br> Deleted: shall be reserved for the LOM <br> XML extension composite XSDs (see <br> Annex B) provided by IEEE. <br> Deleted: NOTE- <br> Deleted: extension elements <br> Deleted: the <br> Deleted: is free to |

### 5.3 Table format and organization

This $\$$ Standard uses tables to describe the requirements for the LOM XML instances. The tables express requirements for each of the LOM data elements in the LOMv1.0 base schema. These requirements are

- LOM data element: The name of the LOM data element in the LOMv1.0 base schema
- XML name: The 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.
- 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: If a LOM data element is not the top-level LOM element, the requirements on the maximum 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 maximum number of times the LOM element may appear in a LOM XML instance. An " $\infty$ " 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 Standard uses three designators for the order: "Ordered", "Unordered", and "Unspecified". "Ordered" 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: binding of

Deleted: ).

Each table that describes an aggregate element includes the direct-descendant subelements of that aggregate element. For example, Table 1 in 5.4 describes the LOM data element and includes the LOM data element's direct-descendant subelements, such as General and Life Cycle. However, the table does not include the subelements of the General element, such as Identifier and Title. If a subelement is also an aggregate element, a separate table describes the subelement with its direct-descendant subelements. If information beyond that provided by IEEE 1484.12.1-2002 is needed to define subelements in a LOM XML instance, that information is provided in subclauses of the aggregate element description.

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 1484.12.12002, the values from IEEE 1484.12.1-2002 shall be used.

Although the Max column may indicate that a subelement may appear in an aggregate element an unbounded maximum number of times, an application that processes a LOM XML instance shall process at least the SPM number of subelements (see 5.1.1).

### 5.4 LOM

Table 1 describes the LOM element and its direct-descendant subelements.
Table 1-The LOM element


| Deleted: first-generation |  |
| :---: | :---: |
| Deleted: first-generation |  |
| Deleted: first-generation |  |
| Deleted: first-generation |  |
| Deleted: name |  |
| Deleted: Data |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Identifier <br> Titleđ <br> Languagef <br> Description ${ }^{\top}$ <br> Keyword $\\|$ <br> Coverageđ <br> Structure <br> Aggregation Level |  |
| Deleted: <lifecycle> |  |
| Deleted: Version ${ }^{-1}$ <br> Status $\boldsymbol{\\|}$ <br> Contribute |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Identifier ${ }^{-1}$ <br> Contribute§ <br> Metadata ${ }^{\\|}$ <br> Schemađ <br> Language |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Format\| <br> Size『 <br> Location $\\|$ <br> Requirement\| <br> Installation $\\|$ <br> Remarks\\| |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Interactivity Type\\| | ... [2] |
| Deleted: Unspecified |  |
| Deleted: n |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Cost\\| | ... [3] |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Kind $\downarrow$ | ... [4] |
| Deleted: n |  |


| LOMdata element | - XML name | Subelements | Min | Max | - Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Annotation | annotation | See 5.4.8 | 0 | $\frac{\infty}{(30)}=$ | Unordered | Unspecified |
| Classification | classification | See 5.4.9 | 0 | 10 <br> (40) | Unordered | Unspecified |

## Namespace declaration

The LOM XML instance shall include a namespace declaration that declares the LOM namespace for the LOM element and its components. The LOM namespace shall be "http://ltsc.ieee.org/xsd/LOM" as defined in 5.2,

The namespace declaration used in conformance with "W3C Recommendation, Namespaces in XML, 1.1." Examples are shown in Figures 1 and 2, respectively
<lom xmlns="http://ltsc.ieee.org/xsd/LOM">
</lom>

Figure 1—An example default namespace declaration

```
<lom:lom xmlns:lom="http://ltsc.ieee.org/xsd/LOM">
. . .
</lom:lom>
```

Figure 2—An example prefix-specific namespace declaration

NOTE-A namespace declaration allows an application to recognize a LOM XML instance as one that contains LOM data elements described in this Standard.

### 5.4.1 General

Table 2 describes the General element and its direct-descendant subelements.


## Deleted: :

Deleted: is required so that applications will
Deleted: LOM data as
Deleted: that conforms to
Deleted: standard
Deleted: first generation

Table 2-The General element

| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| General | Identifier <br> Title <br> Language <br> Description <br> Keyword <br> Coverage <br> Structure | 0 | 1 | Unspecified | Unspecified |  |
| Aggregation Level |  |  |  |  |  |  |


| LOMdata element | XML name | - Subelements | - Min | Max | - Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | , identifier | Catalog | 0 | $\bigcirc$ | Unordered | Unspecified |
|  |  |  |  | (10) |  |  |
| Catalog | catalog | None | 0 | 1 | Unspecified | CharacterString |
| Entry | entry | None | 0 | 1 | Unspecified | CharacterString |

Table 3 describes the Identifier element and its direct-descendant subelements.
Table 3-The Identifier element

| Deleted: name |
| :--- |
| Deleted: Data...* |
| Deleted: <...> |
|  |


| Deleted: name |  |
| :--- | ---: |
| Deleted: Data...* | $\ldots[15]$ |
| Deleted: <...> | $\ldots$ [16] |
| Deleted: Unspecified |  |
| Deleted: n |  |
| Deleted: <...>> | $\ldots[17]$ |
| Deleted: <...> | $\ldots[18]$ |

### 5.4.1.2 Language

## Deleted: the

Deleted: 5.5..1
Deleted: the

## Deleted: source of the

## Deleted: <

Deleted: value
Deleted: be one of

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

- 1
$-2$
$-3$
$-4$


### 5.4.2 Life Cycle

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

| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Life Cycle | lifeCycle | Version <br> Status <br> Contribute | 0 | 1 | Unspecified | Unspecified |



| LOM data element | XML name | - Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Version | version | See 5.4.4 | 0 | 1 | Unspecified | LangString |
| Status | status | See 5.5.5 | 0 | 1 | Unspecified | Vocabulary |
| Contribute | contribute, | See 5.4.2.2 | 0 | $\frac{\infty}{(30)}$ | Ordered | Unspecified |

### 5.4.2.1 Status

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

- draft
- final
- revised
- unavailable


### 5.4.2.2 Contribute

Table 5 describes the Contribute element and its direct-descendant subelements.

Table 5-The Contribute element

| LOMdata element | XML name | - - Subelements | Min | Max | - Order | LOM data type- |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Contribute | contribute | Role <br> Entity <br> Date | 0 | $\infty$ <br> (30) | Ordered | Unspecified |
| Role | role | See 5.5.5 | 0 | 1 | Unspecified | Vocabulary |
| Entity | entity | None | 0 | $\begin{aligned} & -\infty \\ & -\frac{10}{(40)} \end{aligned}$ | Ordered | CharacterString $==$ |
| Date | date | See 5.5.2 | 0 | 1 | Unspecified | DateTime |

### 5.4.2.2.1 Role

If the value space for the source subelement of the Role element is the LOMv1.0 base schema (i.e., <source>LOMv1.0</source> ), then the valid values for the Value subelement of the Role element shall come from the following list of tokens:

- author
- publisher
- unknown

| Deleted: name |  |
| :---: | :---: |
| Deleted: Data |  |
| Deleted: * |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: String |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Sourcef Value |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Roleđ <br> Entity <br> Date |  |
| Deleted: n |  |
| Deleted: *Common data types a ... [19] |  |
| Deleted: source of the |  |
| Deleted: < \ |  |
| Deleted: value |  |
| Deleted: be one of |  |
| Deleted: first-generation |  |
| Deleted: name |  |
| Deleted: Data |  |
| Deleted: * |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: n |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Source¢ $\ldots$ [20] |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: vCard |  |
| Deleted: n |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: DateTime ${ }^{\text {a }}$... [21] |  |
| Deleted: *Common data types a ... [22] |  |
| Deleted: source of the |  |
| Deleted: < \ |  |
| Deleted: value |  |
| Deleted: be one of |  |

- initiator
- terminator
- validator
- editor
- graphical designer
- technical implementer
- content provider
- technical validator
- educational validator
- script writer
- instructional designer
- subject matter expert


### 5.4.2.2.2 Entity

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.

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 de facto standard W3C XML Schema definition language binding for the vCard specification existed.

### 5.4.3 Meta-Metadata

Table 6 describes the Meta-Metadata element and its direct-descendant subelements.
Table 6-The Meta-Metadata element

| LOM data element | -- XML name | - Subelements | Min | Max | Order - - | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Meta-Metadata | metaMetadata | Identifier | 0 | 1 | Unspecified | Unspecified |
|  |  | Contribute |  |  |  |  |
|  |  | Metadata |  |  |  |  |
|  |  | Schema |  |  |  |  |
|  |  | Language |  |  |  |  |
| Identifier | identifier | See 5.4.3.1 | 0 | $\begin{aligned} & \stackrel{\infty}{(10)}== \\ & \underline{(10)} \end{aligned}$ | Unordered $=$ | Unspecifified |
| Contribute | contribute | See 5.4.3.2 | 0 | $\begin{aligned} & \frac{\infty}{(10)}== \\ & \underline{(10)} \end{aligned}$ | Ordered $===$ | Unspecified |
| Metadata Schema | metadataSchema | None |  | $\frac{\infty}{(10)}=$ | Unordered | CharacterString |


| LOM data <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Language | language | None | 0 | 1 | Unspecified | CharacterString |

### 5.4.3.1 Jdentifier

Table 7 describes the Identifier element and its direct-descendant subelements.
Table 7-The Identifier element

| LOMdata element | XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | , identifier | Catalog . . <br> Entry | 0 | $\stackrel{\infty}{(10)}$ | Unordered $==$ | Unspecified |
| Catalog | catalog | None | 0 | 1 | Unspecified | CharacterString |
| Entry | entry | None | 0 |  | Unspecified | CharacterString |

### 5.4.3.2 Contribute

Table 8 describes the Contribute element and its direct-descendant subelements.

Table 8-The Contribute element

| LOM data element | XML name | - Subelements | ${ }^{-}$Min | Max | - Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Contribute | contribute, | Role <br> Entity <br> Date | 0 | $\stackrel{\infty}{(10)}$ | Ordered | Unspecified |
| Role | role | See 5.5.4 | 0 | 1 | Unspecified | Vocabulary |
| Entity | entity | None | 0 | $\begin{aligned} & +\infty \\ & (10) \end{aligned}$ | Ordered | CharacterString |
| Date | date | See 5.5.2 | 0 | 1 | Unspecified | DateTime |

### 5.4.3.2.1 Role

If the value space for Source subelement of the Role element is the LOMv1.0 base schema (i.e., <source>LOMv1. $0</$ source>), then the valid values for the Value subelement of the Role element shall come from the following list of tokens:

$$
\begin{aligned}
& \text { - creator } \\
& \text { - validator }
\end{aligned}
$$

| Deleted: name |  |
| :---: | :---: |
| 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: ${ }^{\text {* }}$ Common data types a ... [24] |  |
| Deleted: first-generation |  |
| Deleted: name |  |
| Deleted: Data |  |
| Deleted: * |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: n |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: Source ${ }^{\text {d }}$ | . [25] |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: vCard |  |
| Deleted: N |  |
| Deleted: < |  |
| Deleted: > |  |
| Deleted: DateTime ${ }^{\text {d }}$. |  |
| Deleted: *Common data types a ... [27] |  |
| Deleted: source of the |  |
| Deleted: < \ |  |
| Deleted: value |  |
| Deleted: be one of |  |

### 5.4.3.2.2 Entity

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.

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 de facto standard W3C XML Schema definition language binding for the vCard specification existed.

### 5.4.3.3 Metadata Schema

For those LOM XML instances that are strictly conforming to this Standard (see 4 ), if the Metadata Schema element (metadataSchema) is present in a LOM XML instance, the element shall contain the value LOMv1.0 i.e., <metadataSchema>LOMv1.0</metadataSchema>)

For those LOM XML jnstances that are conforming but not strictly conforming (i.e., contain extended data model elements, see 4), if the Metadata Schema element (metadataSchema) is present in a LOM XML instance, the element shall contain the value LOMv1.0, and additional occurrences of metadataSchema should list all other metadata schemas or authoritative specifications used to create the LOM XML instance.

### 5.4.3.4 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.

### 5.4.4 Technical

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

Table 9-The Technical element

| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Technical | technical | Lize <br> Location <br> Requirement <br> Installation <br> Remarks <br> Other Platform <br> Requirements <br> Duration | 0 | 1 | Unspecified | Unspecified |


| Deleted: standard |
| :--- |
| Deleted: 4 |
| Deleted: < |
| Deleted: > |
| Deleted: . $\\|$ |
| Deleted: Instances |
| Deleted: 4 |
| Deleted: < |
| Deleted: > |
| Deleted: |
|  |
| Deleted: > shall |

## Deleted: be

Deleted: values

Deleted: first-generation

| Deleted: name |
| :--- |
| Deleted: Data |
| Deleted: $\boldsymbol{*}$ |
| Deleted: $<$ |
| Deleted: $>$ |


| LOM data element | $\ldots$ XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Format | format | None | 0 | $\frac{80}{(40)}=$ | Unordered | CharacterString |
| Size | size | None | 0 | 1 | Unspecified | CharacterString |
| Location | location | None | 0 | $\begin{aligned} & 20 \\ & (10) \end{aligned}$ | Ordered | CharacterString |
| Requirement | requirement | See 5.4.4.3 | 0 | $\begin{aligned} & \infty \\ & (40)= \\ & \underline{(40)}= \end{aligned}$ | Unordered | Unspecified |
| Installation Remarks | installationRemarks | None | 0 | 1. | Unspecified | LangString |
| Other Platform Requirements | otherPlatformRequirements | None | 0 | 1 | Unspecified | LangString |
| Duration | duration | None | 0 | 1. | Unspecified | Duration |


| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Requirement | requirement $\ldots$ | OrComposite | 0 | 0 |  |  |
| OrComposite | orComposite |  | See 5.4.4.3.1 | 0 | $(40)$ |  |

### 5.4.4.1 Format

The value held by the Format element shall be a character string where the value of the character string shall be one of the following:

- A literal that is the canonical lexical representation of a Multipurpose Internet Mail Extension (MIME) type value from RFC 2048; or
- The token non-digital.


### 5.4.4.2 Size

The value held by the Size element shall be a combination of the digits "0". . "9". The Size value in a LOM XML instance shall be a valid non-negative integer as defined by the XML Schema derived data type nonNegativeInteger (see XML Schema, Part 2).

### 5.4.4.3 Requirement

Table 10 describes the Requirement element and its direct-descendant subelement.
Table 10-The Requirement element

### 5.4.4.3.1 OrComposite

Table 11 describes the OrComposite element and its direct-descendant subelements.

Table 11—The OrComposite element

| LOMdata element | XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| OrComposite | prCompositer | _Type <br> Name <br> Minimum Version <br> Maximum Version | 0 | $\frac{\infty}{(40)}=$ | Unordered | Unspecified |
| Type | type, | See 5.5.5 | 0 | 1 | Unspecified | Vocabulary |
| Name | name, | See 5.5.5 | 0 | 1 | Unspecified | Vocabulary |
| Minimum_Version | minimumVersion | None | 0 | 1 | Unspecified | CharacterString |
| Maximum_Version | maximumVersion | None | 0 | 1 | Unspecified | CharacterString |

### 5.4.4.3.1.1 Type

The Type and Name elements shall be present as a pair. If one element is present in a LOM XML instance, the other shall be present, also.

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

- operating system
- browser


### 5.4.4.3.1.2 Name

The Type and Name elements shall be present as a pair. If one element is present in a LOM XML instance, the other shall be present, also.

If the value space for Source subelement of the Name element is the LOMv1.0 base schema (i.e., <source>LOMv1.0</source>), and the value of the Value subelement of the Type element is the token operating system, then the valid values for the Value subelement of the Name element shall come from the following list of tokens:

```
- pc-dos
- ms-windows
- macos
- unix
- multi-os
- none
```

If the value space for the Source subelement of the Name element is the LOMv1.0 base schema (i.e., <source>LOMv1.0</source>), and the value of the Value subelement of the


Deleted: source of the
Deleted: <
Deleted: Type element's

Type element is the token browser, then the valid values for the Value subelement of the Name element shall come from the following list of tokens:

## Deleted: value

Deleted: be one of

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


| LOM data element | - . XML name | - - Subelements | Min | Max | - Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Typical Age Range | typicalAgeRange | See 5.5.4 | 0 | $\frac{p}{(5)}==$ | Unordered $=$ | LangString |
| Difficulty | difficulty | See 5.5.5 | 0 | $1=$ | Unspecified | Vocabulary |
| Typical Learning Time | typicalLearningTimes | See 5.5.3 | 0 | 1 | Unspecified | Duration |
| Description | description | See 5.5.4 | 0 | $\begin{aligned} & \frac{20}{10}= \\ & \underline{(10)} \end{aligned}$ | Unordered $=$ | LangString |
| Language | language | None |  | $\begin{aligned} & 20 \\ & \underline{(10)} \\ & \hline \end{aligned}$ | Unordered | CharacterString |

### 5.4.5.1 Interactivity Type

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

```
- active
- expositive
- mixed
```


### 5.4.5.2 Learning Resource Type

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

- exercise
- simulation
- questionnaire
- diagram
- figure
- graph
- index
- slide
- table
- narrative text
- exam
- experiment

| Deleted: name |
| :---: |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: Unspecified |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: Sourceđ Value |
| Deleted: < |
| Deleted: > |
| Deleted: Duration <br> Description |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: Unspecified |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: n |
| Deleted: *Common data types and data elements are defined in. $\mathbb{I}$ |
| 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 |

- problem statement
- self assessment
- lecture


### 5.4.5.3 Interactivity Level

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

- very low
- low
- medium
- high
- very high


### 5.4.5.4 Semantic Density

If the value space for the Source subelement of the Semantic Density element is the LOMv1.0 base schema (i.e., <source>LOMv1.0</source>), then the valid values for the Value subelement of the Semantic Density element shall come from the following list of tokens:

- very low
- low
- medium
- high
- very high


### 5.4.5.5 Intended End User Role

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

- teacher
- author
- learner
- manager

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

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

Deleted: source of the

## Deleted: <

Deleted: value
Deleted: element shall be one

### 5.4.5.6 Context

If the value space for the Source subelement of the Context element is the LOMv1.0 base schema (i.e., <source>LOMv1. $0</$ source>), then the valid values for the Value subelement of the Context element shall come from the following list of tokens:

- school
- higher education
- training
- other


### 5.4.5.7 Difficulty

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

- very easy
- easy
- medium
- difficult
- very difficult


### 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.

### 5.4.6 Rights

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

| LOM data element | - XML name | Subelements | Min | Max | Order | $\frac{\text { LOM data }}{\text { type }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Rights | rights | Cost | 0 | $1 \ldots$ | Unspecified | Unspecified |
|  |  | Copyright And Other Restrictions |  |  |  |  |
|  |  | Description |  |  |  |  |
| Cost | cost | See 5.5.5 | 0 | $1==$ | Unspecified | Vocabulary |

Deleted: source of the

## Deleted: <

Deleted: value
Deleted: be one of

Deleted: <
Deleted: value
Deleted: be one of

| Deleted: be |
| :--- |
| Deleted: values |
| Deleted: $5.5 . .1$ |
| Deleted: first-generation |

Deleted: first-generation

| Deleted: name |
| :--- |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: and |
| Deleted: < |
| Deleted: > |
| Deleted: Sourcef |
| Value |


| $\begin{aligned} & \text { LOM data } \\ & \text { element } \end{aligned}$ | XML name | Subelements | Min | Max | Order | $\frac{\text { LOM data }}{\text { type }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Copyright And Other Restrictions | copyrightAndOtherRestrictions, | See 5.5.5 = = | 0 | $1=$ | Unspecified | Vocabulary |
| Description | description | See 5.5.4 | 0 | 1 | Unspecified | LangString |

### 5.4.6.1 Cost

If the value space for the Source subelement of the Cost element is the LOMv1.0 base schema (i.e., <source>LOMv1. $0</$ source>), then the valid values for the Value subelement of the Cost element shall come from the following list of tokens:

```
- yes
- no
```


### 5.4.6.2 Copyright And Other Restrictions

If the value space for the Source subelement of the Copyright And Other Restrictions element is the LOMv1.0 base schema (i.e., <source>LOMv1. $0</$ source>), then the valid values for the Value subelement of the Copyright And Other Restrictions element shall come from the following list tokens:

```
- yes
```

- no


### 5.4.7 Relation

Table 14 describes the Relation element and its direct-descendant subelements.
Table 14-The Relation element

| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Relation | relation | Kind <br> Resource | 0 | $\begin{aligned} & \frac{\infty}{(100)} \\ & \frac{1}{2} \end{aligned}$ | Unordered | Unspecified |
| Kind | kind | See 5.5.5 | 0 | 1 | Unspecified | Vocabulary |
| Resource | resource | See 5.4.7.2 | 0 | 1 | Unspecified | Unspecified |

### 5.4.7.1 Kind

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 Kind element shall come from the following list of tokens:

| Deleted: name |
| :--- |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: Sourcef |
| Value |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: ${ }^{*}$ Common data types and data |
| elements are defined in . |
| Deleted: source of the |
| Deleted: < |
| Deleted: value |
| Deleted: be one of |
| Deleted: source of the |
| Deleted: Copy Right |
| Deleted: < |
| Deleted: value |
| Deleted: element shall be one |


| Deleted: first-generation |
| :--- |
| Deleted: name |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: Sourcef |
| Value |
| Deleted: < |
| 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 |


| LOMdata element | - XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Resource | resource | Identifier <br> Description | 0 | 1 | Unspecified | Unspecified |
| Identifier | identifier | See 5.4.7.2.1 | 0 | $\frac{\infty}{(10)}$ | Unordered $=$ | Unspecified |
| Description | description | See 5.5.4 | 0 | $\frac{\infty}{(10)}$ | Unordered $=$ | LangString |

### 5.4.7.2.1 Jdentifier

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

695

| LOMdata element | XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | identifier | Catalog <br> Entry | 0 | $\begin{aligned} & \infty \\ & \underline{(10)} \\ & \hline \end{aligned}$ | Unordered | Unspecified |
| Catalog | catalog | None | 0 | 1 | Unspecified | CharacterString |
| 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: ${ }^{\text {Common data types and data }}$ |
| elements are defined in . |
| Deleted: first-generation |
| Deleted: name |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: Unspecified |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: < |


| LOMdata element | -- XML name | - Subelements | Min | Max | - Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Classification | classification | Purpose <br> Taxon Path <br> Description <br> Keyword | 0 | $\infty$ <br> (40) | Unordered | Unspecified |
| Purpose | purpose | None | 0 | 1 | Unspecified | Vocabulary |
| Taxon Path | , taxonPath | See 5.4.9.2 | 0 | $\frac{\infty}{(15)}$ | Unordered | Unspecified |
| Description | description | See 5.5.4 | 0 | 1 | Unspecified | LangString |
| Keyword | keyword | See 5.5.4 | 0 | $\begin{aligned} & -\infty \\ & \stackrel{\infty}{(40)} \\ & \hline \end{aligned}$ | Ordered | LangString |

Deleted: *Common data types and data elements are defined in. $\mathbb{I}$
Deleted: first-generation

| Deleted: name |
| :---: |
| Deleted: Data |
| Deleted: * |
| Deleted: < |
| Deleted: > |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: vCard |
| Deleted: < |
| Deleted: > |
| Deleted: DateTime <br> 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: Sourceđ <br> Taxon |
| Deleted: Undefined |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: n |

### 5.4.9.1 Purpose

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

- discipline
- idea
- prerequisite
- educational objective
- accessibility restrictions
- educational level
- skill level
- security level
- competency


### 5.4.9.2 Taxon Path

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

| LOM data element | XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Taxon Path | taxonPath | Source <br> Taxon | 0 | $\frac{-20}{(15)}$ | Ordered | Unspecified |
| Source | source, | See 5.5.4 | 0 | 1 | Unspecified | LangString |
| Taxon | taxon | See 5.4.9.2.1 | 0 | $\frac{\infty}{(15)}$ | Ordered | Unspecified |

### 5.4.9.2.1 Taxon

Table 20 describes the Taxon element and its direct-descendant subelements.
Table 20-The Taxon element

| LOMdata element | - XML name | -- Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Taxon | taxon | Id <br> Entry | 0 | $\frac{\infty}{(15)}=$ | Ordered | Unspecified |
| Id | , id | None | 0 | 1 | Unspecified | CharacterString |
| Entry | entry | See 5.5.4 | 0 | 1 | Unspecified | LangString |

Deleted: *Common data types and data elements are defined in. 1

## Deleted: source of the

## Deleted: <

## Deleted: value

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: first-generation |
|  | Deleted: name |
|  | Deleted: Data |
|  | Deleted: * |
|  | Deleted: < |
|  | Deleted: > |
|  | Deleted: n |
|  | Deleted: < |
|  | Deleted: > |
|  | Deleted: < |
|  | Deleted: > |
|  | Deleted: String |

### 5.5 Common data types and elements

Data types that are used in the LOM XML $\underline{\text { Schema }}$ binding are defined in $5.5 .1=5.5 .5$.

### 5.5.1 CharacterString

For those elements defined in this Standard that have a LOM data type of CharacterString, the W3C XML Schema definition language binding shall have an XML Schema primitive data type of string. The string data type supports the repertoire of ISO/IEC 10646-1. The 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."

### 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:

- DateTime
- Description

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

Table 21 describes the DateTime data type and its direct-descendant subelements.
Table 21-The DateTime data type

| LOM data <br> element | XML name | $\underline{\text { Subelements }}$ | $\underline{\text { Min }}$ | $\underline{\underline{\text { Max }}}$ | $\underline{\text { Order }}$ | $\underline{\text { LOM data type }}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| $\underline{\text { DateTime }}$ | - | DateTime <br> Description | $\underline{0}$ | $\underline{1}$ | $\underline{\text { Unspecified }}$ | $\underline{\text { Unspecified }}$ |
| DateTime | $\underline{\text { dateTime }}$ | $\underline{\text { None }}$ | $\underline{0}$ | $\underline{1}$ | $\underline{\text { Unspecified }}$ | CharacterString |
| $\underline{\text { Description }}$ | $\underline{\text { description }}$ | $\underline{\text { See } 5.53 .2}$ | $\underline{0}$ | $\underline{1}$ | $\underline{\text { Unspecified }}$ | $\underline{\text { LangString }}$ |

NOTE-The DateTime data type is made up of two elements, DateTime and Description, that are 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 5.4.2.2).

### 5.5.2.1 DateTime

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

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

Deleted: 1.1.1

## Deleted: standard

Deleted: be
Deleted: xs:
Deleted: xs:
Deleted: -
Deleted: xs:
Deleted: <xs :
Deleted: > construct

Deleted: e data type is represented as an XML element. The DateTime data type may contain the following subelements: $\boldsymbol{\pi}$ <\#>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 Timeๆ
Table 21 describes the Date Time element and its first-generation subelements. Table 21-The DateTime element $\|$
LOM name
... [28]

YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]]] where:

- YYYY; The 4-digit year ( $\geq 0001$ )
- MM: The 2-digit month (01 through 12 where $01=$ January, etc.)
- DD: The 2-digit day of month ( 01 through 31, depending on the values of month and year)
- hh; Two digits of hour (00 through 23) (am and pm are NOT allowed)
- mm; Two digits of minute (00 through 59)
$-\quad \mathrm{ss}_{\mathrm{\gamma}} ;$ Two digits of second (00 through 59)
- s ; One or more digits representing a decimal fraction of a second
- TZD; The time zone designator ( $Z$, for coordinated universal time [UTC] or $+\mathrm{hh}_{2}$ -hh, $+\mathrm{hh}: \mathrm{mm}_{2}$ or $-\mathrm{hh}: \mathrm{mm}_{2}$ where hh is two digits of hour and mm is two digits of minute)

If the DateTime element contains a value, at least the four-digit year shall be present. If additional parts of the date and time are included, the character literals " - ", T, ": ", and ". " are part of the character lexical representation for the DateTime element.

If the time portion is present, but the time zone designator is not present, the time zone is interpreted as being UTC.

## NOTES:

1-Because of restrictions placed on the value space, the W3C XML Schema definition language primitive data type dateTime could not be used to create conforming LOM XML instances. Therefore, a regular expression was created to represent all valid dates and times supported by IEEE 1484.12.1-2002.

2-The date portion represents dates in the Common Era (CE), only. The date portion follows the Gregorian calendar for dates from October 15, 1582, forward, and the Julian calendar for dates prior to October 15, 1582, independent of locale. Dates Before Common Era (BCE) and other cases should be represented using the Description element (see 5.5.2).

3 -The square bracket meta characters $([$ and $\sqrt{ }])$ indicate optional elements that may appear zero or one time in the character lexical representation of the DateTime. These meta characters do not appear in the result; only the associated values described appear (e.g., DD is replaced by the corresponding 2 -digit value for day of month).

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

### 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:

## - Duration

- Description

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

### 5.5.3 Duration

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

| LOM data <br> element | -- XML name | Subelements | Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Duration | V | Duration | 0 | 1 | Unspecified | Unspecified |
|  |  | Description |  |  |  |  |
| Duration | duration | None | 0 | 1 | Unspecified | CharacterString |
| Description | description | See 5.5.3.2 | 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).

### 5.5.3.1 Duration

The value space for the Duration element is a pattern that shall be defined according to the following constraints:
$P[y Y][m M][d D][T[h H][n M][s[. s] S]]$ where:
$-y_{i}$ The number of years (integer, $\geq 0$ )

- $m_{;}$The number of months (integer, $\geq 0$, not restricted, e.g., $>12$ is acceptable)
- d; The number of days (integer, $\geq 0$, not restricted, e.g., $>31$ is acceptable)
$-h_{i}$ The number of hours (integer, $\geq 0$, not restricted, e.g., $\geq 23$ is acceptable)
$-\eta^{2}$ The number of minutes (integer, $\geq 0$, not restricted, e.g., $>59$ is acceptable)
- $s_{i}$ The number of seconds or fraction of seconds (integer, $\geq 0$, not restricted, e.g., $>59$ is acceptable)

The character literal designators $P_{\infty} Y_{p} M_{\infty} D_{\infty} T_{p} H_{2} M_{2}$ and $S_{z}$ shall appear if the corresponding nonzero value is present.

If the Duration element contains a value, the designator $P$ shall be present. If the value of years, months, days, hours, minutes or seconds is zero, the value and corresponding designation (e.g., M) may be omitted, but at least one designator with a positive value shall be present

| Deleted: . |
| :--- |
| Deleted: element |
| Deleted: first-generation |
| Deleted: element |
| Deleted: name |
| Deleted: Data |
| Deleted: unspecified |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: String |
| Deleted: which |
| Deleted: rules |
| Deleted: is the |
| Deleted: > |
| Deleted: , not restricted |
| Deleted: is the |
| Deleted: > |
| Deleted: . The |
| Deleted: is the |
| Deleted: and value must always be |
| Dresent. The designator "P" is always |
| Deleted: ", " " |
| Deleted: > |
| Deleteted: " ", " ", " " " |
| Deleted: is the |
| Deleted: > |
| Deleted: is the |
| Deleted: > |
| Deleted: is the |
| Deleted: > |
| Deleted: " " |
|  |

in addition to the designator $P$. The designator $T_{v}$ shall be omitted if all of the time (hours, minutes and seconds) is zero.

Negative durations are not supported.
NOTES:
1-Because of the restrictions placed on the value space, the W3C XML Schema definition language primitive data type dateTime could not be used to create conforming LOM XML instances. A regular expression was built to represent all valid date/times supported by IEEE 1484.12.1-2002.

2-The value is designated in the Gregorian calendar.
3-The ordering of durations may be indeterminate (e.g., 1 month may be $28,29,30$, or 31 days).
$4-$ The square bracket meta characters ([ and $]$ ) 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 corresponding value for the number of days in the duration and is followed by the character literal designator $\mathrm{p}_{\mathrm{p}}$ ).

5-This value space is based on ISO 8601:2000 [B2].

### 5.5.3.2 Description

The Description data type is represented as a LangString data type (see 5.5.4).

### 5.5.4 LangString

The LangString data type is represented as a set of XML elements. LOM data elements with a data type of LangString may contain the following subelements:

- String
- Language

If present, the subelements listed above shall be contained by the element for which the LangString data type is defined. The order of appearance of subelements shall not be significant. Within a LOM XML instance, the LOM data element Language shall be specified as an attribute of LOM data elements that are of type LangString.

Table 23 describes the LangString data type and its, direct-descendant subelements.

## Table 23-The LangString data type

| $\begin{aligned} & \text { LOM data } \\ & \text { element } \end{aligned}$ | - XML name | - Subelements | - Min | Max | Order | LOM data type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| LangString | E | String <br> Language | 0 | $\infty$ <br> (10) | Unordered | Unspecified |
| String | , string, | None | 0 | 1 | Unspecified | CharacterString |
| Language | language | None | 0 | 1 | Unspecified | CharacterString |

NOTE-The LangString data type is made up of two elements, String and Language, that are direct descendents of all aggregate elements that are defined to be of type LangString. For example, String and Language are the direct descendents of the aggregate element Title (see 5.4.1).

### 5.5.4.1 Language

The Language element shall be a character string consisting of a required language code followed by multiple, optional, hyphen-prefixed subcodes.

The following constraints apply to the language code part of the character string:

- $\quad 2$-letter codes are defined in ISO 639-1.
- 3-letter codes are defined in ISO 639-2.
- The 1 -letter code " $i$ " is reserved and uses as a prefix for registrations defined by the Internet Assigned Numbers Authority (IANA).
- The 1 -letter code "x" is reserved and used as a prefix for private use.

The following constraints apply to the first subcode part of the character string:

- 2-letter subcodes are ISO 31661 alpha-2 country codes.
- Subcodes from 3 to 8 in length letters are registered with IANA.

Constraints for additional subcodes are unspecified.
The value held by the character string shall be a valid language code as defined by the XML Schema derived data type language (see XML Schema, Part 2).

ISO 639-2 specifies two code sets, one for bibliographic applications (ISO 639-2/B) and one for terminology applications (ISO 639-2/T). Either code set may be used.

## NOTES:

1-The language code is normally given in lower case and the subcodes (if any) in upper case. However, the values are case insensitive.

2-The XML Schema derived data type language does not enforce all the constraints on the language code that are described above.

Deleted: element

| Deleted: name |
| :---: |
| Deleted: Data |
| Deleted: Unspecified |
| Deleted: n |
| Deleted: < |
| Deleted: > |
| Deleted: * |
| Deleted: *language is an XML attribute, not an XML element. ${ }^{\\|}$ <br> The Language element shall be a character string consisting of a required language code followed by multiple, optional, hyphen-prefixed subcodes (see examples below). The Language element is defined in |
| Deleted: .\\| |
| Deleted: valid LanguageID where\\| LanguageID = Langcode (""Subcode) * $\\|$ <br> The following rules apply to the Langcode part of the |
| Deleted: : |
| Deleted: Two |
| Deleted: Three |
| Deleted: value |
| Deleted: value |
| Deleted: rules |
| Deleted: Subcode |
| Deleted: Two |
| Deleted: of |
| Deleted: Rules |
| Deleted: NOTE-The Lang |

Deleted: . $\|$
Deleted: valid LanguageID where ${ }^{\|}$ LanguageID = Langcode ("Subcode)*の
The following rules apply to the Langcode part of the

Deleted: Two

## Deleted: Three

Deleted: value
Deleted: rules
Deleted: Subcode
Deleted: Two
Deleted: of

Deleted: NOTE-The Lang

## Examples

"en-GB"
"de"
"fr-CA"
"it"
"i-bnn" (IANA Bunun)

### 5.5.5 Vocabulary

The Vocabulary data type is represented as a set of XML elements. LOM data elements with a data type of Vocabulary may contain the following subelements:

- Source
- Value

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

Table 23 describes the Vocabulary data type and its direct-descendant subelements.

Table 24-The Vocabulary data type

| LOMdata <br> element | XML name | Subelements | Min | Max | Order | LOM data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Vocabulary | L_ | Source <br> Value | Unspecified | Unspecified | Unspecified | Unspecified |
| Source | source | None | 0 | 1 | Unspecified | CharacterString |
| Value | value | None | 0 | 1 | Unspecified | CharacterString |

NOTE-The Vocabulary data type is made up of two elements, Source and Value, that are direct descendents of all aggregate elements that are defined to be of type Vocabulary. For example, Source and Value are the direct descendents of the aggregate element Aggregation Level (see 5.4.1.4).

### 5.5.5.1 Value

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).

NOTE-The XML Schema derived data type token supports the repertoire of ISO/IEC 106461 . The token data type ensures the support of multiple languages including multibyte languages. Multibyte languages are not used by the LOMv1.0 base schema vocabularies but may be used if the source of a vocabulary is not LOMV1.0.

Deleted: en-US-philadelphia"

## Deleted: <\#>Token $\|$

For those elements defined in this standard that have a data type of Token, the W3C XML Schema definition language binding shall be an xs : token. The xs: token data type supports the repertoire of ISO/IEC 10646-1. The
xs: token data type ensures the support of multiple languages including multibyte languages. $\pi$
NOTE-The <xs : token> construct is defined in "XML Schema Part 2."匹 <\#>vCard $\ddagger$
The vCard data type is represented as a CharacterString data type. The canonical lexical representation of the vCard value shall be a valid vCard[] as defined in IETF RFC 2426:1998. $\dagger$ 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 de facto standard W3C XML Schema definition language binding for the vCard specification existed.\|

Deleted: The vocabulary data type is represented as an XML element. Table 24 describes the Vocabulary element and its first-generation subelements. ${ }^{\|}$
Table 23-The Vocabulary element 1
Deleted: name
Deleted: Data

| Deleted: Unspecified |
| :--- |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: Token |

Deleted: Token

## 894 <br> Annex A

(informative)

## Bibliography

[B1] IEEE $100^{\mathrm{TM}}$, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.
[B2] ISO 8601:2000, Data elements and interchange formats - Information interchange Representation of dates and times.
[B3] ISO/IEC 11404:1996, Information Technology_Programming Languages, their Environments and System Software Interfaces-Language-Independent Datatypes.
[B4] W3C Recommendation (04 February 2004), Extensible Markup Language (XML) 1.1.

Deleted: A1

Deleted: [A2] IETF RFC 2425:1998, MIME Content-Type for Directory Information. $\|$
Deleted: A3
Deleted: A4

Deleted: [A5] W3C Recommendation
24 October 2001, XML Information Set. $\|$

## Annex B

(informative)

## Internet availability of XSD files

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

The XSD files are available at the following URL:

- http://ltsc.ieee.org/xsd/lomv1.0/

NOTE-This Standard does not require the use of an XSD to validate a LOM XML instance.

## 914 Annex C

(informative)

## XSD file descriptions

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.

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.

When building XSDs, implementers can decide when to enforce LOMv1.0 base schema constraints. These constraints can be defined in an XSD with the intent that schema processors will perform validation of the constraints or that validation will be performed by processing. Clause 5 defines the criteria to create valid LOM instances; XSD implementers must decide how to develop XSD schema files according to these constraints. Regardless of the implementation, a LOM instance must conform to the rules defined in Clause 5PSVI.

This Standard includes informative W3C XML XSDs for implementers to satisfy many of the requirements of IEEE 1484.12.1-2002 (see Annex B) and the practices of its existing adopters, The XSDs are intended to be used by XML processing tools that support the W3C XML Schema definition language (see XML Schema, Parts 1 and 2). The LOMv1.0 base schema places requirements on a LOM XML instance some of which cannot be expressed in W3C XML Schema definition language. These requirements are defined in Clause 5 of this Standard and the LOMv1.0 base schema.

The goal is to provide alternative XSDs, each of which supports the LOM XML Schema binding by making certain choices about which binding constraints to describe in the XSDs. Constraints defined in an XSD should be enforced by tools that support XML Schema validation. Constraints not expressed in an XSD are enforced by other means, such as processing the PSVI. No matter where constraint enforcement is performed, a conforming or strictly conforming LOM XML instance has to conform to all constraints defined in Clause 5 of this Standard and the LOMv1.0 base schema.

A predefined, composite XSD (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 extension use. Three additional predefined composite XSDs are provided that enforce different validation constraints (see C.1.2 - C.1.3). The four composite XSDs draw from several sets of component XSDs (see C.2).

The need for alternative XSDs was motivated by the variety of ways in which application developers need to use the XSDs. The approach of providing alternative XSDs addresses the

## Deleted: concrete

Deleted: where

## Deleted: defined in the XSD,

Deleted: outside the XSD
Deleted: in some implementation-
defined, post-schema

Deleted: standard

## Deleted: (see Annex B)

Deleted: A predefined, composite XSD (see C.1.1) is provided that can be tailored in multiple ways for particular preferences about LOM element ordering, checking vocabulary values, and using extensions. Three additional predefined composite XSDs are provided that enforce different validation rules (see C.1.2 - C.1.4). The four composite XSDs draw from several sets of component XSDs (see C.2).9

Deleted: rules
Deleted: Rules
Deleted: Rules
Deleted: the rule
Deleted: done
Deleted: must
Deleted: rules
needs of the various adopting communities to produce conforming or strictly conforming LOM XML instances consistent with the communities' processing and usage requirements.

The presence or absence of the enforcement of any part of the LOM XML Schema binding by a particular XSD in no way relieves a particular LOM XML instance from satisfying all of the LOM XML Schema binding constraints in order to be considered a conforming or strictly conforming LOM XML instance. The presence of alternative XSDs merely provides developers choices in determining how to enforce the LOM XML Schema binding constraints described in this Standard.

## C. 1 Composite XSDs

The lom. xsd composite XSD acts as a template after which the other composite XSDs (lomLoose.xsd, lomStrict.xsd, and lomCustom.xsd) have been modeled. Each of these XSDs is structured from component XSDs described in C.2. By default the lom.xsd composite XSD encodes a particular set of binding constraints in the W3C XML Schema definition language such that a LOM XML instance will strictly conform to IEEE 1484-12.1-2002. Each of the individual composite XSDs defined in C.1.1 - C.1.3 are particular combinations of W3C XML Schema definition language.

As shown in Figure C.1, the lom. xsd composite XSD is an abstract composite XSD from which three additional composite XSDs (lomLoose.xsd, lomStrict.xsd and lomCustom. xsd) have been built. These composite XSDs model certain validation approaches based on the selection of a combination of multiple component XSDs. (See C.1.1 C.1.3 for more information about these composite XSDs.)


Figure C.1-The relationship of the composite XSDs.

NOTE-The names of XSDs in C.1.1 - C.1.3, lomLoose.xsd lomStrict. xsd, and lomCustom. xsd, refer to files that are accessed through URLs (see Annex B).

Deleted: their

## Deleted: rules

Deleted: rules
Deleted: standard

Deleted: Composite XSDs are defined in C.1.1-C.1.4.

## Deleted: defined

Deleted: NOTE-The specific names of XSDs in C.1.1-C.1.4, lom . xsd, lomLoose.xsd,
lomStrict.xsd, and lomCus-
tom. xsd, refer to files that are accessed through URLs (see Annex B). $\|$ C.1.1.1 Generic approach $\uparrow$

The lom. xsd composite XSD encodes a particular set of binding rules in the W3C XML Schema definition language. The lom. xsd composite XSD.
<\#>Defines a mechanism to describe custom vocabularies (source/value pairs) in addition to the vocabularies defined in the LOMv1.0 base schema. A validating parser will check that vocabulary values in a LOM instance conform to either the vocabularies expressed in the LOMv1.0 base schema or those expressed in the custom vocabulary; $\boldsymbol{\|}$
<\#>Expresses the uniqueness constraints defined by the LOMv1.0 base schema; and $\|$
<\#>Defines a mechanism to describe extensions to the LOMv1.0 base schema Extensions must be defined in a set of supporting XSDs. A validating parser will check that the extensions used in a LOM instance conform to rules defined in the supporting XSD. .
C.1.1.2 Loose composite XSD ๆ The lomLoose. xsd composite XSD encodes a particular set of binding rules in the W3C XML Schema definition language. The

Deleted: composite XSD

## C.1.1 lomStrict.xsd

$\checkmark$
The lomstrict. xsd composite XSD encodes a particular set of binding constraints in the W3C XML Schema definition language such that a LOM XML instance will strictly conform to IEEE 1484-12.1-2002. The lomStrict . xsd composite XSD

- Limits vocabularies (source/value pairs) to those defined in Clause 5 of this Standard;
- Expresses the uniqueness constraints defined in Clause 5 of this Standard; and
- Does not permit extensions to the LOMv1.0 base schema (see 5.1.3).


## C.1.2 IomCustom.xsd

The lomCustom. xsd composite XSD encodes a particular set of binding constraints in the W3C XML Schema definition language such that a LOM XML instance will conform to IEEE 1484-12.1-2002. The lomCustom. xsd composite XSD

- Defines a mechanism to describe vocabularies (source/value pairs) in addition to those defined in Clause 5 of this Standard. A validating parser will check that vocabulary values in a LOM XML instance conform to either the vocabularies defined in Clause 5 or those defined in an extended vocabulary;
- Expresses the uniqueness constraints defined in Clause 5 of this Standard; and
- Defines a mechanism to describe extensions to the LOMv1.0 base schema (see 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 constraints defined in the supporting XSDs that include extensions.


## C.1.3 IomLoose.xsd

The lomLoose.xsd composite XSD encodes a particular set of binding constraints in the W3C XML Schema definition language such that a LOM XML instance may not either conform or strictly conform to IEEE 1484-12.1-2002. The lomLoose . xsd composite XSD

- Does not provide validation to any defined vocabulary values (i.e., values defined in Clause 5 of this Standard or values defined in an extended vocabulary). A validating parser will verify that the vocabulary values are valid tokens, only;
- Does not express the uniqueness constraints defined in Clause 5 of this Standard; and
- Defines a mechanism to describe extensions to the LOMv1.0 base schema (see 5.1.3). Extensions are defined in a set of supporting XSDs. A validating parser will check that the extensions used in a LOM XML instance conform to constraints defined in the supporting XSDs that include extensions.

NOTE-The absence of the validation of vocabulary values does not relieve a particular LOM XML instance from satisfying the vocabulary requirements defined in Clause 5 of this Standard.

Deleted: <\#>Does not provide validation to any defined vocabulary schema (i.e., the LOMv1.0 base schema or a custom schema). A validating parser will verify that the vocabularies are valid tokens, only; ${ }^{6}$
Does not express the uniqueness
Deleted: defined by the LOMv1.0 base schema; and
<\#>Defines a mechanism to describe extensions to the LOMv1.0 base schema. Extensions must be defined in a set of supporting XSDs. A validating parser will check that the extensions used in a LOM instance conform to rules defined in the supporting XSDs that include extensions. ${ }^{\|}$
NOTE-The absence of the enforcement of vocabulary values does not relieve a particular LOM instance from satisfying vocabulary requirements defined in the LOMv1.0 base schema. Applications that require the use of vocab/loose.xsd component XSD should enforce those vocabulary requirements by other means. ${ }^{\|}$ C.1.1.3 Strict composite XSD I The lomStrict.xsd composite XSD encodes a particular set of binding rules

Deleted: the LOMv1.0 base schema, only
Deleted: by LOMv1.0 base schema
Deleted: does
Deleted: any
Deleted: C.1.1.4 Custom composite XSD

Deleted: rules

Deleted: will conform to IEEE 1484 12.1-2002. The lomCustom. xsd composite XSD
Defines a mechanism to describe custom vocabularies (source/value pairs) in addition to the vocabularies defined in the LOMv1.0 base schema. A validating parser will check that vocabulary values in a LOM instance conform to
Deleted: the vocabularies expressed in the LOMv1.0 base schema or those expressed in the custom vocabulary;
Deleted: Expresses
Deleted: by the LOMv1.0 base schema; and
Defines a mechanism to describe extensions to the LOMv1.0 base schema. Extensions must be defined in a set of supporting XSDs. A validating parser will check that the extensions used in a LOM instance conform to rules defined

Applications that require the use of vocab/loose.xsd component XSD should enforce those vocabulary requirements by other means.

## 1014 C. 2 Component XSDs

A component XSD represents a constituent of a composite XSD. The following component XSDs are used by the composite XSDs defined in C.1.

- common/anyElement.xsd: Defines the base XML Schema model groups for elements and attributes used for extended data elements and XML attributes.
- common/dataTypes.xsd; Defines global data type declarations for data types defined in Clause 5 of this Standard.
- , common/elementNames.xsd; Defines global element declarations for each of the data elements defined in Clause 5 of this Standard. This component XSD is used to check for the uniqueness of data elements declared to be unique within their aggregate elements by the presence of the uniqueElementName attribute. The XML Schema constraint unique is used to enforce uniqueness constraints,
- common/elementTypes.xsd: Defines global data type declarations for data elements defined in Clause 5 of this Standard. This component XSD defines the aggregation relationships among the LOM data elements. These aggregation relationships enforce the LOMv1.0 base schema requirement that elements can be present in a LOM XML instance as elements of the aggregate element to which they belong, only.
- common/rootElement.xsd: Defines the element name declaration for the element that contains all other LOM data elements for a LOM XML instance.
- common/vocabTypes.xsd; Defines global data type declarations for those LOM data elements that have values taken from a vocabulary data type defined Clause 5 of this Standard.
- common/vocabValues.xsd; Defines the standard vocabulary value declarations as defined in Clause 5 of this Standard. These declarations are used in conjunction with both vocab/custom.xsd and vocab/strict.xsd,
- extend/custom.xsd: Defines the XML Schema model groups customElements and customAttributes to support validation of extended data elements and attributes. The model groups in this component XSD reference the model groups defined in common/anyElement. xsd. This component XSD should be used if extensions are to be supported in LOM XML instances (see 5.1.3). Note: Assuming proper processing of the PSVI, LOM XML instances that use the extend/custom.xsd component XSD and extensions will conform but not strictly conform to this Standard.
- extend/strict.xsdi Defines the XML Schema model groups customElements and customAttributes. The model groups are defined as empty model groups and are used with other component XSDs to support the

Deleted: or part element
Deleted: The
Deleted: component XSD defines
Deleted: schema
Deleted: IEEE 1484.12.1-2002
Deleted: The
Deleted: component XSD defines
Deleted: IEEE 1484.12.1-2002.
Deleted: parent
Deleted: element
Deleted: multiplicity
Deleted: of no more than one item within a LOM instance

## Deleted: The

Deleted: component XSD defines
Deleted: schema
Deleted: IEEE 1484.12.1-2002.
Deleted: relationship
Deleted: LOMv1.0 base schema
Deleted: IEEE 1484.12.1-2002
Deleted: only
Deleted: The
Deleted: component XSD defines
Deleted: in IEEE 1484.12.1-2002
Deleted: The
Deleted: component XSD defines
Deleted: IEEE 1484.12.1-2002.
Deleted: vocabulary value
Deleted:
Deleted: <\#> The com-
mon/rootElement.xsd component XSD defines the element name declaration for the element that contains all other elements for a LOM instance. $\mathbb{}$ \|
<\#>The common/anyElement.xsd component XSD defines a global declaration that is re-used for extension data elements. $\|$
<\#>The unique/loose.xsd component XSD defines attribute group declarations for data elements to support the schema-based validation of uniqueness constraints within a LOM instance where the exact set of attributes associa ... [29]
Deleted: The
Deleted: component XSD defines
Deleted: content
Deleted: strict
validation of strictly conforming LOM XML instances. This component XSD 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/strict.xsd component XSD and, therefore, not supporting extensions, 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 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 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.
- 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.
- 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.
- 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. 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 instance 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.

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 vo-
cab/custom.xsd enforces strict
Deleted: custom
Deleted: IEEE 1484.12.1-2002 or from a custom token set.
Deleted: The vocab/strict.xsd component schema enforces strict conformance with the standard vocabulary values specified in the LOMv1.0 base schema, 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. $\uparrow$

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: $\uparrow$
NOTES: $\mathbb{\|}$
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

- vocab/strict.xsd: Enforces that vocabulary values are only those specified in Clause 5 of this Standard. This component XSD doēs nō allow extended vōabulary values from other sources. This component XSD supports the validation of strictly conforming vocabulary values by checking that both sources and values are from a token set defined in Clause 5.

Deleted: C. 3 Enabling extensions $\|$ With certain constraints, the LOMv1.0 base schema anticipates the need to introduce data elements not included in the standard collection of elements. These constraints aređ
<\#>Extensions retain the original value spaces and data types of LOMv1.0 base schema data elements; and $\mathbb{}$ ब
<\#>Extensions provide additional elements 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" processContents="lax"/>).ब
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. $\|$ NOTES:『
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 extensions. $\uparrow$
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.

## Annex D

(informative)

## Enabling extended data elements and attributes

The LOMv1.0 base schema anticipates the need to introduce extensions not included in the standard collection of elements (see 5.1.3). These extensions have to meet the following constraints:

- Extensions to the LOMVv1.0 base schema retain the value spaces and data types of LOM data elements from the LOMv1.0 base schema; and
- Extensions do not define data types or value spaces for aggregate elements in the LOMv1.0 base schema.

These requirements imply that if an aggregate element contains extensions, the extensions do not redefine the intended meaning (i.e., value space and data type) of the aggregate element. To meet these requirements, the informative schemas (see Annex B) include an any element in the content model for aggregate elements (<xs:any namespace="\#\#other" processContents="lax"/>).

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 LOMv1.0 base schema 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 Schema processor will validate any well-formed LOM XML instance.

NOTES:
1 -To ensure that conforming LOM XML instances are created, an organization that develops extensions also provide an XSD that defines the validation constraints for the extensions.

2-Organizations providing extensions may choose to define namespaces for these extensions. If an organization defines a namespace, then the XSDs provided by this Standard may have to be altered to support the namespace.

## D. 1 Enabling extended data elements

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.

To enable extended data elements, the extend/custom.xsd component XSD should be used in the construction of the composite XSD. This component XSD defines the XML Schema model group customElements that references the customElements model group
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.

The relevant component XSDs and how they enable extended data elements are described below.

- common/anyElement.xsd: Defines the base XML Schema model group for elements used for extended data elements.
- common/dataTypes.xsd and common/elementTypes.xsd: For each aggregate complexType declaration defined in this component XSD, an XML Schema model group reference to the customElements declaration is defined.
- common/vocabTypes.xsd: For each vocabulary complexType declaration defined in this component XSD, an XML Schema model group reference to the customElements declaration is defined.
- extend/custom.xsd: Defines the XML Schema model group customElements to support validation of extended data elements. The XML Schema model group in this component XSD references the model group defined in the common/anyElement. xsd component XSD.

NOTE-Extensions may be disabled by using the extend/strict. xsd composite XSD.

## D. 2 Enabling extended attributes

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.

To enable extended attributes, the extend/custom.xsd component XSD should be used in the construction of the composite XSD. This component XSD defines the XML Schema model group customAttributes. This model group defines the ability to add any attribute from a namespace other than the namespaces defined in 5.2. The inclusion of this component XSD by a composite XSD enables the validation of extended attributes.

The relevant component XSDs and how they enable extended attributes are described below.

- common/anyElement.xsd: Defines the base XML Schema model group for elements used for extended data elements.
- common/dataTypes.xsd and common/elementTypes.xsd: For each aggregate complexType declaration defined in this component XSD, an XML Schema model group reference to the customAttributes declaration is defined.
- common/vocabTypes.xsd: For each vocabulary and value complexType declaration defined in this component XSD, an XML Schema attributeGroup

1168
1169
1170
1171
1172
reference to the customAttributes XML Schema model group declaration is defined.

- extend/custom.xsd: Defines the XML Schema model group customAttributes to support validation of extended attributes.

NOTE-Extensions may be disabled by using the extend/strict.xsd composite XSD.

## Annex E

## (informative)

## XSD implementation choices

Major implementation choices contained in the example XSDs (see Annexes B and C) are discussed in E. 1 -E. 6.

## E. 1 Data types

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 Clause 5 of this Standard. 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.
The following global W3C XML Schema definition language types are defined:

- CharacterString: an alias for the XML Schema primitive data type string.
- LanguageId: an alias for the XML Schema derived data type language
- VCard: an alias for CharacterString.
- MimeType: an alias for CharacterString.
- Size: an alias for the XML Schema derived data type nonNegativeInteger.
- LanguageString: a sequence of zero or more string elements with an optionally defined language attribute. The language attribute describes the language of the value held by the string element.
- DateTime: optional subelements dateTime and description,
- Duration: optional subelements duration and description,

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).

The よanguageString 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.

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: $\mathbb{\\|}$ <br> <\#>LanguageIdNone: an alias for xs:token. $\\|$ <br> LanguageIdOrNone: a union of the LanguageID and LanguageIdNone. |
| Deleted: xs: |
| Deleted: LangString |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: VCard and MimeType are data types that act as placeholders. Thes data types may be further defined in future editions of this standard. $\\|$ |
| Deleted: LangString |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: LangString. |

dateTime subelement contains a restricted pattern of characters. The pattern, defined using a regular expression in W3C XML Schema definition language, is

$$
\begin{aligned}
& ([0-9]\{3\}[1-9] \mid \text { [0-9]\{2\}[1-9][0-9] | }[0-9][1-9][0-9]\{2\} \mid \text { | } 1- \\
& 9][0-9]\{3\})(\backslash-(0[1-9] \mid 1[0-2])(\backslash-(0[1-9]|[1-2][0-9]| 3[0- \\
& 1])(\mathrm{T}([0-1][0-9] \mid 2[0-3])(:[0-5][0-9](:[0-5][0-9](\backslash .[0-9]\{1,\}(Z \\
& \mid((\backslash+\mid \backslash-)([0-1][0-9] \mid 2[0-3]):[0-5][0-9])) ?) ?) ?) ?) ?) ?) ?
\end{aligned}
$$

The restricted character string pattern is used instead of the XML Schema primitive data type because of the restrictions placed on the value space defined in IEEE 1484,12.1-2002. These restrictions cannot be enforced by the XML Schema primitive data type dateTime.

The Duration data type has the optional subelements duration of type CharacterString, and description, of type LanguageString. The CharacterString type for the duration subelement contains a restricted pattern of characters. The pattern, defined using a regular expression in W3C XML Schema definition language, is

$$
\begin{aligned}
& P([0-9]\{1,\} Y)\{0,1\}([0-9]\{1,\} M)\{0,1\}([0-9]\{1,\} D)\{0,1\}(\mathrm{T}([0- \\
& 9]\{1,\} H)\{0,1\}([0-9]\{1,\} M)\{0,1\}([0-9]\{1,\}(\backslash .[0- \\
& 9]\{1,\})\{0,1\} S)\{0,1\})\{0,1\}
\end{aligned}
$$

The restricted character string pattern is used instead of the XML Schema primitive data type because of the restrictions placed on the value space defined in IEEE 1484,12.1-2002. These restrictions cannot be enforced by the XML Schema primitive data type duration XML Schema data type.

## E. 2 Elements

The common/elementNames.xsd component XSD defines a collection of global W3C XML Schema definition language data 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.

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.

For each LOM data element, an XML element name is assigned by the convention of using camel-case capitalization of the full name of the LOM data element with an initial lower-case letter_(lowerCamelCase) (see 5.4 and 1.1).

Each LOM data element is given a global W3C XML Schema definition language element declaration, type declaration, and attribute group declaration. The XML names defined in Clause 5 are used for the names of each of these declarations.


| Deleted: spelling |
| :--- |
| Deleted: provided with |
| Deleted: a global W3C XML Schema <br> definition language |
| Deleted: a global W3C XML Schema <br> definition language |
| Deleted: Schema element name is |
| Deleted: name |

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.

## E. 3 Aggregates

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
a) An ordering restriction is not defined in IEEE 1484.12.1-2002. Each element should allow arbitrary ordering of its subelements.
b) Each element should enforce multiplicity constraints defined in IEEE 1484.12.12002 on its subelements.
c) The aggregation structure should be preserved without the introduction of additional 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).
Requirements (a) and (b) above introduce complexity in expressing the content model for aggregate elements. Requirement (c) above 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 additional container elements (c).

Three alternatives for the selection of the content model for aggregate elements could be implemented.

NOTE-In E.3.1 - E.3.5, "a", "b", and "c" in parentheses or square brackets refer to the requirements listed above.

## E.3.1 Using the XML Schema element sequence,

The use of the XML Schema element sequence provides a way to meet the requirements of multiplicity (b) and avoiding additional container elements (c) but does not provide arbitrary ordering (a). If subelements are required to appear in a specific order, the XSD can easily enforce the multiplicity constraints without introducing additional container elements. Figure E. 1 shows how the sequence element can be used to define the LOMv1.0 base schema.

```
<xs:complexType name="lom">
    <xs:sequence>
        <xs:element minOccurs="0" ref="general"/>
```

```
    <xs:element minOccurs="0" ref="lifeCycle"/>
    <xs:element minOccurs="0" ref="metaMetadata"/>
    <xs:element minOccurs="0" ref="technical"/>
    <xs:element minOccurs="0" maxOccurs="unbounded"
        ref="educational"/>
        <xs:element minOccurs="0" ref="rights"/>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="relation"/>
        <xs:element minOccurs="0" maxOccurs="unbounded"
        ref="annotation"/>
            <xs:element minOccurs="0" maxOccurs="unbounded"
            ref="classification"/>
        </xs:sequence>
</xs:complexType>
```

Figure E.1-An example using sequence construct

According to Figure E. 1 the general element can appear zero or one time (satisfies [b]). The example does not introduce additional container elements (satisfies [c]). However, the general element, if present, has to precede the lifeCycle element (does not satisfy [a]).

LOM XML instances created using this approach may be non-conforming.

## E.3.2 Using the XML Schema element all

The XML Schema element all provides a way to meet the requirements of arbitrary ordering (a). Using the all element requires the introduction of additional container elements (c) to meet the requirement of multiplicity (b). The all element requires the introduction of additional container elements to collect multiple instances of a given element. The LOMv1.0 base schema contains elements that can have multiplicities of more than one (e.g., Educational). Figure E. 2 illustrates how the all element can be used to define the LOMv1.0 base schema.

```
<xs:complexType name="lom">
    <xs:all>
        <xs:element minOccurs="0" ref="general"/>
        <xs:element minOccurs="0" ref="lifeCycle"/>
        <xs:element minOccurs="0" ref="metaMetadata"/>
        <xs:element minOccurs="0" ref="technical"/>
        <xs:element minOccurs="0" ref="educationals"/> <!-- container -->
        <xs:element minOccurs="0" ref="rights"/>
        <xs:element minOccurs="0" ref="relations"/> <!-- container -->
        <xs:element minOccurs="0" ref="annotations"/> <!-- container -->
        <xs:element minOccurs="0" ref="classifications"/>
            <!-- container -->
    </xs:all>
</xs:complexType>
```

Figure E.2-An example using the all element

Deleted: D
Deleted: <xs:
Deleted: >

Deleted: artificial
Deleted: must
Deleted: lifecycle
Deleted: D
Deleted: [xs:all](xs:all) II
The use of
Deleted: <xs :
Deleted: > construct defines
Deleted: ) and multiplicity (b). However
Deleted: <xs :
Deleted: > construct introduces
Deleted: use
Deleted: artificial
Deleted: XML <xs :
Deleted: > construct
Deleted: artificial
Deleted: D
Deleted: XML sequence construct

Deleted: >

Deleted: D
Deleted: <xs:
Deleted: >

According to Figure E. 2 the general element can appear zero or one time (satisfies [b]) anywhere in the sequence of elements (satisfies [a]). However, the example introduces additional container elements, such as the educationals element (does not satisfy [c]), where the plural educationals indicates zero or more educătional elements.

## LOM XML instances created using this approach may be non-conforming.

## E.3.3 Using the XML Schema element choice

The use of the XML Schema element choice provides a way to meet the requirements of arbitrary ordering (a) and avoiding additional containers (c) but does not enforce multiplicity constraints (b) By allowing any subelement to appear any number of times, the W3C XML Schema definition language types can easily be constructed to allow arbitrary ordering of the subelements without having to introduce additional container elements. Figure E. 3 shows how the choice element can be used to define the LOMv1.0 base schema.

```
<xs:complexType name="lom">
```

    <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="general"/>
            <xs:element ref="lifeCycle"/>
            <xs:element ref="metaMetadata"/>
            <xs:element ref="technical"/>
            <xs:element ref="educational"/>
            <xs:element ref="rights"/>
            <xs:element ref="relation"/>
            <xs:element ref="annotation"/>
            <xs:element ref="classification"/>
    </xs:choice>
    </xs:complexType>

## Figure E.3-An example using the choice element

The example in Figure E. 3 does not introduce additional container elements (satisfies [c]) and allows arbitrary ordering (satisfies [a]). However, all of the elements may appear any number of times (does not satisfy [b]).

This solution accepts a superset of what the LOMv1.0 base schema allows without introducing additional complexity in the LOM XML instances or the LOM XSD. For applications willing to enforce LOM data element multiplicity constraints on the PSVI, this may represent a sufficient solution.

1359

Deleted: artificial
Deleted: (
Deleted: D

Deleted: <xs:
Deleted: >

Deleted: construct defines
Deleted: artificial
Deleted: A more flexible solution can be found by temporarily giving up on the enforcement of the multiplicity constraints.

Deleted: artificial
Deleted: D
Deleted: XML sequence construct
<xs:
Deleted: >

## Deleted: D

Deleted: <xs:
Deleted: >

Deleted: artificial

Deleted: uniqueness
Deleted: post-schema validation XML infoset

Deleted: D

## E.3.4 Using the XML Schema element choice with the XML

 Schema constraint uniqueThe use of the XML Schema element choice with the XML Schema constraint unique provides a way to meet the requirements of arbitrary ordering (a) and avoiding additional containers (c) and enforces multiplicity constraints (b). Using this approach, the multiplicity constraints can be enforced using the XML Schema constraint unique. For each element with a multiplicity of at most one in its aggregate element, define an attribute called uniqueElementName with a fixed default value equal to the name of the element, and use this attribute as the element key for unique constraint. Figure E. 4 shows how the unique constraint can be used to enforce multiplicity constraints for element uniqueness when the LOMv1.0 base schema defines that an element cannot appear more than one time.

```
<xs:element name="lom" type="lom">
    <xs:unique name="lomUnique">
        <xs:selector xpath="*"/>
        <xs:field xpath="@uniqueElementName"/>
    </xs:unique>
    </xs:element>
```

Figure E.4-An example using the unique constraint

One such constraint using the unique constraint is needed for each aggregate element that contains subelements with multiplicities of at most one.

The disadvantages of this approach are the introduction of the uniqueElementName attribute and the introduction of additional complexity in the XSD.

This solution maximizes the amount of validation that can be done by a parser. For applications that cannot process the PSVI, the addition of the uniqueElementName attribute may be a sufficient solution.

This solution supports the validation of both conforming and strictly conforming LOM XML instances.

## E.3.5 Summary

Subclauses E.3.1 - E. 3.4 defined alternative approaches to aggregate elements. A summary of those alternatives is given below
$\mid-$ sequence; Does not satisfy requirement (a), which states that the subelements appear in an arbitrary order. This alternative does not conform to IEEE 1484.12.12002.


- all: Does not satisfy requirement (c), which preserves the aggregation structure "defined in IEEE 1484.12.14--2002. The use of the XML Schema element all requires the introduction of arbitrary container elements (e.g., educationals). This alternative requires all subelements of the container to be collected under the aggregate element.
- choice; Does not enforce uniqueness constraints. All of the elements may appear any number of times. This alternative does not satisfy requirement (b) and does not conform to IEEE 1484.12.1-2002. This alternative is defined in the unique/loose. xsd component XSD (see C.2).
- choice with unique: Overcomes the problems with the XML Schema element choice, An attribute is applied to each element with a multiplicity of at most one. The attribute is enforced by applying an identity constraint. This alternative is defined in the unique/strict.xsd component XSD (see C.2).


## E. 4 Vocabularies

The selection of the content model for LOM data elements of type Vocabulary involves satisfying three requirements that arise either directly or indirectly from the LOMv1.0 base schema and from how vocabularies are to be used by applications. These are
a) Data elements of type Vocabulary should allow arbitrary values (source/value pairs).
b) If the source of a vocabulary value is the LOMv1,0 base schema (i.e., <source>LOMv1. $0</$ source>), then the token used to represent the vocabulary value is defined in Clause 5 of this Standard.
c) If the source of a vocabulary value is not the LOMv1, 0 base schema (i.e., not <source>LOMv1. $0</$ source>), then the token used to represent the vocabulary value should not conflict with any of the tokens defined in Clause 5 of this Standard.

The W3C XML Schema definition language cannot easily accommodate the goal of strict type checking for vocabulary values in a one-pass model It can provide validation of a fixed set of vocabularies, but cannot validate arbitrary vocabularies. In LOM XML instances, the fixed vocabularies are defined based on the source of the vocabulary, If the source of the vocabulary is defined, W3C XML Schema definition language constructs can be defined to aid in the validation of the vocabularies,
Three alternatives for expressing LOM vocabularies in W3C XML Schema definition language are presented in E.4.1 - E.4.3. These three alternatives provide options on the validation approaches taken by XML Schema processors.

NOTE-In E.4.1 - E.4.3, " a ", " b ", and " c " in parentheses refer to the requirements listed above.


## E.4.1 vocab/loose.xsd

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

```
<xs:simpleType name="difficulty">
    <xs:restriction base="lom:CharacterString"/>
</xs:simpleType>
```

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

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

This approach accepts a superset of what the LOMv1.0 base schema allows without introducing additional complexity in LOM XML instances, The disadvantage of this approach is that it does not support validation of vocabulary values. Using the vocab/loose. xsd component XSD requires processing the PSVI to validate vocabulary values used in both strictly conforming and conforming LOM XML instances.

## E. 4.2 vocab/strict.xsd

If vocabularies are limited to only those defined in the LOMv1.0 base schema, then the vocab/strict.xsd component XSD may be used. This component XSD defines constraints for a validation of LOMv1.0 base schema vocabularies. The use of this component 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 and c). This approach is a straightforward implementation of the values allowed by the vocabulary elements. As shown in Figure E.6, each element that takes its values from a vocabulary is defined where the source element is given a fixed value of LOMv1.0, and the value of the value element is derived from an enumerated list of the appropriate vocabulary values.

```
<xs:simpleType name="difficulty">
    <xs:union memberTypes="lom:difficultyValues"/>
</xs:simpleType>
<xs:complexType name="difficultyVocab">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="source" type="sourceValue"/>
```

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

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
Deleted: a strict vocabulary validation approach
Deleted: 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: >

```
            <xs:element name="value" type="difficultyValue"/>
            <xs:group ref="ex:customElements"/>
        </xs:choice>
            <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>
<xs:simpleType name="difficultyValues">
        <xs:restriction base="xs:token">
            <xs:enumeration value="very easy"/>
            <xs:enumeration value="easy"/>
            <xs:enumeration value="medium"/>
            <xs:enumeration value="difficult"/>
            <xs:enumeration value="very difficult"/>
        </xs:restriction>
    </xs:simpleType>
```

        Figure E.6-An example from the vocab/strict.xsd component XSD
    Both the source and value elements are optional and may appear in any order. Values are validated against the vocabulary tokens defined in Clause 5 of this Standard for the LOMv1.0 base schema.

This approach accepts a subset of what the LOMv1.0 base schema would allow by excluding vocabulary values with sources other than the LOMv1.0 base schema. The advantage of this approach is that it can validate the vocabulary values defined in this Standard.

## E. 4.3 vocab/custom.xsd

To use both vocabularies that are defined in the LOMv1.0 base schema and extended vocabularies, the vocab/custom. xsd component XSD may be used. This component XSD defines constraints for validation of both LOMv1.0 base schema vocabularies and vocabularies that are defined in other schemas. The use of this component XSD provides a way to meet the goal of allowing values from arbitrary sources (a) by supporting the validation of both vocabulary values from other sources (c) and LOMv1.0 base schema vocabulary values (b). This approach provides additional flexibility in the partial validation of both standard and extended vocabulary values. As shown in Figure E. 7 each element that takes its values from a vocabulary is defined where the source element is allowed to have an arbitrary value, and the value of the value element is derived from a custom list of the appropriate vocabulary values.

```
<xs:simpleType name="difficulty">
    <xs:union memberTypes="lom:difficultyValues lx:difficultyValues"/>
</xs:simpleType>
<xs:complexType name="difficulty">
    <xs:complexContent>
        <xs:extension base="difficultyVocab">
            <xs:attributeGroup ref="ag:difficulty"/>
        </xs:extension>
    </xs:complexContent>
```

| Deleted: D |
| :--- |
| Deleted: using |
| Deleted: < |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: , |
| Deleted: standard |
| Deleted: standard |
| Deleted: the XSD |
| Deleted: D |
| Deleted: If |
| Deleted: limited to both those |
| Deleted: with |
| Deleted: a custom defined set of |
| Deleted: n a custom vocabulary valida- |
| tion approach may be desirable. The |
| Deleted: schema |
| Deleted: rules |
| Deleted: by |
| Deleted: the vocab/custom . xsd |
| Deleted: schema |
| Deleted: but does not meet the require- |
| ments of arbitrary ordering (b |
| Deleted: avoiding artificial containers |
| (c). |
| Deleted: supplies |
| Deleted: < < custom |
| Deleted: D |
|  |

```
</xs:complexType>
<xs:simpleType name="difficultyValues">
    <xs:restriction base="xs:token">
        <xs:enumeration value="very easy"/>
        <xs:enumeration value="easy"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="difficult"/>
        <xs:enumeration value="very difficult"/>
    </xs:restriction>
</xs:simpleType>
```

Figure E.7-An example from the vocab/custom.xsd component XSD

Both the source and value elements are optional and may appear in any order. Values are validated against the extended vocabulary enumerations.

This approach accepts a superset of what the LOMv1.0 base schema allows and allows the use of both standard and extended vocabulary values. The advantage of this approach is that it retains the ability for the validation of standard vocabulary values while accommodating the use of extended vocabulary values. Although the example in Figure E. 7 restricts vocabulary values to those defined in the LOMv1.0 base schema when the source is LOMv1.0 (i.e., <source>LOMv1. $0</$ source>), the example allows the use of vocabulary values defined in the LOMv1.0 base schema along with extended vocabulary values if the source is not LOMv1. 0.

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 namespace that defines the extended vocabularies. The vocab/custom.xsd component XSD provides a placeholder for this extension (xmlns:lx="http://ltsc.ieee.org/XSD/LOM/custom"). The organization may change this declaration.

## E. 5 Additional notes

This subclause discusses additional decisions that were made to enforce conformance to the LOMv1.0 base schema. The following information applies to implementation choices that were used in this Standard and in the XSD files provided with this Standard (see Annex B). These implementation choices are included in the dataTypes. xsd component XSD.

- Certain elements are defined in IEEE 1484.12.1-2002 to be represented as valid vCard syntax. At the time of publication of this Standard, no de facto standard W3C XML Schema definition language binding for the vCard specification was available. XSD implementations may want to define a type to encapsulate elements of type VCard. The VCard data type does not validate that the associated value represents a well-formed Internet Mail Consortium 3.0 vCard (see IETF RFC 2426:1998).


| Deleted: < |
| :--- |
| Deleted: > |
| Deleted: < |
| Deleted: > |
| Deleted: custom |
| Deleted: custom the |
| Deleted: presence of custom |
| Deleted: However, |
| Deleted: does not restrict |
| Deleted: vocabulary values |
| Deleted: < |
| Deleted: >). Nor does |
| Deleted: prevent |
| Deleted: standard |
| Deleted: may |
| Deleted: schema |
| Deleted: D |

Deleted: Some
Deleted: additional notes apply
Deleted: :

Deleted: standard, there was

Deleted: and define a
Deleted: for those elements that have a value of a vCard.

Deleted: vCard
Deleted: vCard

- 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]).
- 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 language,
- 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.
- The global <description> element has content model LangString and multiplicity 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.
- 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.
- 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: (indicating
Deleted: )

Deleted: The global <entity> element

Deleted: VCard

Deleted: The enumerated values for
Deleted: are listed as dependent on the value
Deleted: corresponding Type
Deleted: the OrComposite element,
Field Code Changed
Deleted: ). The values of the <name> element in the XSD are given as the union of the listed values. 9 A local element declaration for the Source element (see the Taxon path element,
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.

| 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 |  |  |
| 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 Lewis | 11/5/ 2004 2:09:00 PM |
| < |  |  |
| 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 |
| $<$ |  |  |
| 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: [18] Deleted | Scott Lewis | 11/5/2004 2:09:00 PM |
| < |  |  |
| Page 11: [18] Deleted | Scott Lewis | 11/5/2004 2:09:00 PM |
| $>$ |  |  |
| Page 13: [19] Deleted | Scott Lewis | 11/ 5/ 2004 2:28:00 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

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 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.

## Table 21—The DateTime element

| LOM name | XML name | Subelements | Min | Max | Order | Data type |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Date | <date> | DateTime <br> Description | 0 | 1 | Unspecified | DateTime |
| DateTime | <dateTime> | None | 0 | 1 | Unspecified | CharacterString |
| Description | <description> | String | 0 | 1 | Unspecified | LangString |

## 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/rootElement.xsd component XSD defines the element name declaration for the element that contains all other elements for a LOM instance.
The common/anyElement.xsd component XSD defines a global declaration that is re-used for extension data elements.
The unique/loose.xsd component XSD defines attribute group declarations for data elements to support the schema-based validation of uniqueness constraints within a LOM instance where the exact set of attributes associated with each element has to be precisely as specified by the LOM XML binding (i.e., where extra, artificial attributes must be avoided). This component XSD is used to relax the enforcement of the uniqueness constraints to avoid the introduction of the uniqueElementName artificial attribute (see D.3.3).
The unique/strict.xsd component XSD defines attribute group declarations for data elements defined in IEEE 1484.12.1-2002 to support schema-based validation of the uniqueness constraints with a LOM instance by introducing the artificial attribute uniqueElementName for each data element that appears with a multiplicity of at most one (see D.3.3).
The extend/custom.xsd component XSD defines the content model group customElements to support validation of custom data elements.
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.
$\qquad$


[^0]:    $\mid{ }^{1}$ For information on normative references, see Clause 2.

[^1]:    ${ }^{1}$ The numbers in brackets correspond to those of the bibliography in Annex A.

