學習物件詮釋資料之可延伸標示語言架構
定義語言繫結 — 中文草案

中華民國國家標準 CNS	學習物件詮釋資料之可延伸標示語言架構定義語言繫結	總號.. xxxxx-x
		類號.. xxxx-x

Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata

目錄

1. 適用範圍

本標準之範圍與目標於第1.1及1.2節中討論

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

1.1 範圍

本標準定義全球資訊網協會(World Wide Web Consortium，以下簡稱W3C)的延伸標示語言(Extensible Markup Language，以下簡稱XML)結構，並限定XML1.1文件用以描述IEEE 1484.12.1 TM-2002標準定義之學習物件詮釋資料(learning object Metadata，以下簡稱LOM)實例。本標準定義XML1.1文件中的W3C XML架構定義(XML Schema definition以下簡稱XSD)語言之結構和限制。本標準所有實作均遵循IEEE 1484.12.1-2002標準之定義。

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 Std 1484.12.1TM-2002.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 Std 1484.12.1-2002.

公布日期 年..月..日 經濟部標準檢驗局印行 修訂公布日期 年..月..日

1.2 目標

本標準之目標係允許以XML建立互運LOM實例。本標準採用W3C XML架構定義語言編碼，以允許LOM XML實例能於不同系統間互運性及交換。

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

2. 用語釋義

各項定義與首字母縮寫分別於第2.1與2.2節中說明

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

2.1 定義

本標準應用下述專有名詞及定義，其餘未定義者請參考 The Authoritative Dictionary of IEEE Standards Terms [B1]。

For the purposes of this Standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms* [B1] should be referenced for terms not defined in this clause.

2.1.1 聚合元件：包含其他稱爲子元件之LOM資料元件的LOM資料元件。
參照：子元件。

aggregate element: A LOM data element that contains other LOM data elements called subelements.*See also:* subelement.

2.1.2 組件XSD (Extensible Markup Language Schema definition , XSD)：定義合成架構構件之XSD。
參照：合成XSD。

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.

2.1.3 合成XSD(composite XSD) ：由特異組件XSD所組成的結構謂之合成XSD。
參照：組件XSD。

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.

2.1.4 內容模型：識別特定模型之構成(makeup)(即資料型式、多樣限制、排序)的框架。

content model:A framework that identifies the makeup (i.e., data types, multiplicity constraints,ordering) of a specific model.

2.1.5 資料型式：特異值之性質，指出該值一般功能與加諸該值之運算。

data type:A property of distinct values that indicates the common features of those values and operations on those values.

2.1.6 延伸資料元件：LOMv1.0基礎架構中未定義的資料元件。
參照：LOMv1.0基礎架構。

extended data element:A data element that is not defined in the LOMv1.0 base schema. *See also:*LOMv1.0 base schema.

2.1.7 延伸詞彙：於LOMv1.0基礎架構中，未於型式詞彙(Vocabulary)的LOM資料元件中定義的值域(value space)。
參照：LOMv1.0基礎架構。

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

2.1.8 XML資訊集(XML Infoset)：其他需要參考完整XML文件(W3C, XML資訊集)規格中所使用的一致定義的一種抽象的資料集。
參照：後架構確認資料集。

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 wellformed XML document (W3C, XML Information Set). *See also:* post-schema-validation infoset.

2.1.9 XML架構繫結(XML Schema binding)：資料模型元件在W3C XML架構定義語言中，其行爲、屬性及值域的文字表示法。

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.

2.1.10 學習物件：係 指任何可用作學習、教育或訓練的數位化或非數位化個體。

learning object:Any entity, digital or nondigital, that may be used for learning, education, or training. (IEEE 1484.12.1-2002)

2.1.11 LOM資料元件(LOM data element)：IEEE 1484.12.1標準中所定義包括名字、解釋、大小、排序、值域以及資料型式的資料元件。
參照：LOMv1.0基礎架構。

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 Std 1484.12.1-2002. *See also:* LOMv1.0 base schema.

2.1.12 LOM XML實例(LOM XML instance)：符合IEEE 1484.12.1-2002標準之學習物件的大量詮釋資料集，以XML表示且遵循XML架構繫結的要求和限制。

learning object metadata Extensible Markup Language instance (LOM XML instance):A collection of metadata for a learning object that conforms to IEEE Std 1484.12.1-2002, that is represented in XML, and that adheres to the requirements and constraints of the XML Schema binding.

2.1.13 LOMv1.0基礎架構：於IEEE Std 1484.12.1-2002第6節中定義的大量結構之標準資料項，包含其資料型式、多樣性和容器/組件之間關聯性。

LOMv1.0 base schema:A structured collection of standard data items, including their data types, multiplicities, and container/component relationships, as defined in IEEE Std 1484.12.1-2002, Clause 6.

2.1.14 混合式內容：可含字元資料亦能隨意散置子元件之混合式內容的元件型式(W3C, Extensible Markup Language [XML] 1.1)。

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)

2.1.15 多用途網際網路郵件延伸型式(MIME 型式)：網際網路上分類內容型式的標準方法。

Multipurpose Internet Mail Extensions type (MIME type):A standard way of classifying content types on the Internet.

2.1.16 後架構確認資訊集(post-schema-validation infoset)：符合W3C XML架構處理器所產生文件的XML資訊集合轉換版本。W3C XML架構定義了資料元件在確認原始文件時，加諸XML資訊於其上以產生後架構確認資訊集。

參照：延伸標示語言資訊集合。

post-schema-validation infoset:A transformed version of the XML infoset of a document produced by a conforming 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.

2.1.17 子元件：LOM資料元件包含於其他LOM元件中的稱爲聚合元件。子元件可包含其他子元件，也可以是聚合元件。

參照：聚合元件。

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.

2.1.18 符記：指字元串，其值域不包含換行(#xA)或tab標籤(#x9)字元之一組字串，且無前導或尾端空格(#x20)以及內部兩個以上連續的空格。詞彙空間則是既不包括換行(#xA)、也不包括tab標籤(#x9)字元的一組字串，並無前導或尾端空格(#x20)以及內部兩個以上連續的空格。符記的基本型式是正規字串(normalizedString)，符記有大小寫之分。(Adapted from W3C, XML Schema Part 2).

參照：符記集。

token:A character string. The value space of a token is the set of strings that do not contain the line feed (#xA) or tab (#x9) characters, that have no leading or trailing spaces (#x20), and that have no internal sequences of two or more spaces. The lexical space is the set of strings that does not contain the line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20), and that have no internal sequences of two or more spaces. The base type is normalizedString. Tokens are case sensitive. (Adapted from W3C, XML Schema Part 2). *See also:* token set.

2.1.19 符記集：該集合中每個符記皆爲唯一。

參照：符記。

token set:A set of tokens in which each token is unique. *See also:* token.

2.1.20 唯一性限制：在LOM資料元件之限制強制遵守LOM資料元件於LOM XML實例中爲獨一無二的。假若LOM資料元件具有獨一性的限制，其於LOM XML實例中不會出現或僅出現1次。

參照：學習物件詮釋資料資料元件。

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.

2.1.21 值域：給定資料型式之一組值。(ISO/IEC 11404:1996)

value space:The set of values for a given data type. (ISO/IEC 11404:1996)

備考：IEEE 1484.12.1標準中，值域是典型地全部列舉，或是引用自其他標準或規格。

NOTE—In IEEE Std 1484.12.1-2002, a value space is typically enumerated outright or defined by reference to another standard or specification.

2.2 縮寫

IANA 網際網路地址指派機構(Internet Assigned Numbers Authority)

LOM 學習物件詮釋資料(LOM Learning Object Metadata)

MIME 多用途網際網路郵件延伸標準(MIME Multipurpose Internet Mail Extensions)

PSVI 後架構確認資訊集(PSVI post-schema-validation infoset)

SPM 最小允許上限值(SPM smallest permitted maximum)

UTC 世界標準時(UTC coordinated universal time)

W3C 全球資訊網協會(W3C World Wide Web Consortium)

XML 可延伸標示語言(XML Extensible Markup Language)

XSD XML架構定義(XSD XML Schema definition)

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

3. 參考標準

以下是本標準應用的必要參考文件。有日期的文獻，僅引版本爲準；無日期的文獻則列出含任何修正的最新版參考文件。

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 or corrigenda) applies.

IEEE Std 1484.12.1TM-2002, IEEE Standard for Learning Object Metadata.
IETF RFC 2048:1996, Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures.
IETF RFC 2426:1998, vCard MIME Directory Profile.
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 (28 October 2004), XML Schema Part 1: Structures, Second Edition.
W3C Recommendation (28 October 2004), XML Schema Part 2: Datatypes, Second Edition.
W3C Recommendation (4 February 2004), Namespaces in XML 1.1.

4. 符合性

本標準之符合性於第4.1與4.2節中討論。第4.1與4.2節完全符合LOM XML實例，並符合LOM XML實例在做任何處理前所參考用的詮釋資料。本標準中，「應」被解釋爲指實作的必須要條件；「不應」被解釋爲禁止。

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 before 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.

4.1 完全符合之LOM XML 實例

(1) 應完全符合IEEE 1484.12.1-2002標準所定義之LOM實例。

(2) 應符合第5節的要求。

(3) 不應包括第5節中未定義的詞彙值。

(4) 不應包括第5節中未定義的XML元件或屬性。

(5) 不應包括混合式內容。

A *strictly conforming* LOM XML instance

— Shall be a strictly conforming LOM instance as defined in IEEE Std 1484.12.1-2002

— Shall conform to the requirements of Clause 5

— Shall not include vocabulary values that are not defined in Clause 5

— Shall not include XML elements or attributes that are not defined in Clause 5

— Shall not include mixed content

4.2 符合之LOM XML實例

(1) 應符合IEEE 1484.12.1-2002標準所定義之LOM實例。

(2) 應符合第5節中的要求。

(3) 可包括第5節中未定義的詞彙值。

(4) 可包括第5節中未定義並以第5.1.3節中描述之延伸機制的XML元件或屬性。

(5) 可包括混合式內容。

A *conforming* LOM XML instance

— Shall be a conforming LOM instance as defined in IEEE Std 1484.12.1-2002

— Shall conform to the requirements of Clause 5

— May include vocabulary values that are not defined in Clause 5

— May include XML elements and attributes that are not defined in Clause 5 by using the extension mechanism described in 5.1.3

— May include mixed content

5. LOM XML架構繫結定義

LOM XML架構繫結定義於第5.1節至5.5節中。第5.4節列出LOMv1.0基礎架構定義之LOM資料元件。每個LOM資料元件均有以XML架構繫結的相對應XML元件描述。

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

5.1 一般資訊

如同IEEE1484.12.1-2002標準之定義，所有LOM資料元件皆爲選項的。LOMv1.0基礎架構之XSD不應要求任何LOM資料出現於LOM XML實例中。LOMv1.0基礎架構界定資料項間的聚合關聯性，LOM之XSD於其子元件間定義聚合關聯性以維持LOMv1.0基礎架構中定義的關聯性。LOM

XML實例中，子元件應僅於聚合元件中的聚合關聯性中出現。舉例來說，第5.4.3.1節中，子元件識別符(Identifier)乃定義爲第5.4.3節中描述的詮釋-詮釋資料(Meta-Metadata)元件之一部份。此子元件的呈現自動的意指子元件所屬的聚合元件之呈現。LOMv1.0基礎架構並未定義除聚合關聯性外之LOM資料元件任何序列。因此，LOMv1.0基礎架構的XML架構繫結不應定義除聚合關聯性外，LOM XML實例中元件的任何序列。

備考：本標準的XML命名原則源於序連LOM資料元件命名並採用*lowerCamelCase*大寫的LOM資料元件命名原則。

As defined in IEEE Std 1484.12.1-2002, all LOM data elements are optional. An 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 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 最小允許上限值

W3C XML架構定義語言不支援IEEE1484.12.1-2002標準所定義的SPM概念。W3C XML架構定義語言提供限制最大長度的方法(即字元串的maxLength)，和最多出現次數(即聚合元件的maxOccurs)。然而，此限制不相容於SPM之定義。如果LOM XML實例包含超過LOM資料元件的SPM出現次數，則應用應處理不少於LOM資料元件的SPM出現次數。如果LOM XML實例包含超過字元串中的SPM字元數，則應用應處理不少於字元串中的SPM字元數。(IEEE 1484.12.1-2002標準對LOM資料元件及字元串的SPM值有所定義)。

備考：爲促進互運性，意圖確認符合IEEE 1484.12.1-2002標準之LOM XML實例的XSD建立者，不應使用maxOccurs或maxLength限制或應設定範圍限制爲未設限。

The W3C XML Schema definition language does not support the concept of SPMs as defined in IEEE Std 1484.12.1-2002. The W3C XML Schema definition language does provide a means for restricting maximum lengths (i.e., maxLength for character strings) and maximum numbers of occurrences (i.e., maxOccurs for 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 Std 1484.12.1-2002.)

NOTE—To encourage interoperability, creators of XSDs that intend to validate LOM XML instances against IEEE Std 1484.12.1-2002 should not use maxOccurs or maxLength restrictions or should set the restricted limits to unbounded.

5.1.2 LOM XML實例的符合性限制

與IEEE 1484.12.1-2002有關之符合或完全符合的LOM XML實例，其之效度限制條件於第5.4節和第5.5節中說明。所有符合和完全符合的LOM XML實例應滿足此限制條件。

備考：有些應用程式可能會使用XSD確認LOM XML實例。若一個應用程式用XSD確認LOM XML實例，LOM XML實例的PSVI並不足以決定與IEEE 1484.12.1-2002標準相關的LOM XML實例之效度。使用XSD確認LOM XML實例並不會解除在第5.4節及第5.5節中出現的執行限制的應用程式需求。

Constraints on the validity of a *conforming* or *strictly conforming* LOM XML instance with respect to IEEE Std 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 PSVI of a LOM XML instance is insufficient to determine the validity of the LOM XML instance with respect to IEEE Std 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 延伸的支援

本標準允許LOMv1.0基礎架構有所延伸，正如IEEE 1484.12.1-2002標準的定義，LOMv1.0基礎架構之延伸應保留LOMv1.0基礎架構中LOM資料元件的資料型式及其值域，且不應定義LOMv1.0基礎架構中聚合元件的資料型式及其值域。

本標準允許下列符合但不完全符合之LOM XML實例的延伸型式：

(1) LOMv1.0基礎架構之XML元件延伸：LOMv1.0基礎架構之資料元件可被延伸，此延伸是在第5.2節定義之名稱空間以外的新XML元件。延伸資料元件可加入於任一與聚合元件同義的LOM資料元件。延伸資料元件不應加入於非聚合元件的LOM資料元件。(參照第5.4節及第5.5節)

(2) LOMv1.0基礎的架構之XML屬性延伸：LOM資料元件可由LOM資料元件的屬性定義而延伸，此延伸是在第5.2節定義之名稱空間以外的XML屬性，且可以加入於任何的LOM資料元件。(參照第5.4節及第5.5節)

(3) LOMv1.0基礎架構詞彙資料型式延伸：LOMv1.0基礎架構詞彙係可延伸，此延伸是型式詞彙(Vocabulary)之LOM資料元件所額外增加的符記(參照第5.4節)。若延伸詞彙，則新增詞彙的符記來源應有所識別且不應是LOMv1.0。

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

— *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 5.5).

— *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名稱空間

第5.4節及第5.5節中所有定義的LOM資料元件應於http://ltsc.ieee.org/xsd/LOM名稱空間中定義，此名稱空間由本標準所保留且不應用以表示任何其他的資料元件(參照W3C Recommendation, Namespaces in XML 1.1.)，本名稱空間應不應用以定義LOM XML實例之延伸。下述本標準所提供之名稱空間係用於XSD中(參照附錄B)：

(1) http://ltsc.ieee.org/xsd/LOM/custom
(2) http://ltsc.ieee.org/xsd/LOM/unique
(3) http://ltsc.ieee.org/xsd/LOM/vocab
(4) http://ltsc.ieee.org/xsd/LOM/extend

如果任何組織提供LOM XML實例之延伸，則該組織必須定義該項延伸之名稱空間。

All 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 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.

5.3 表格格式與組織

本標準採用表格描述LOM XML實例之需求，該表格表達LOMv1.0基礎架構中各項LOM資料元件的需求，此需求的說明如下所述：

(1) LOM資料元件：LOMv1.0基礎架構中之LOM資料元件的名稱。

(2) XML名稱：在LOM XML實例中所使用的名稱必須遵循本標準並與LOM資料元件一致。在第5.5節中的表格使用破折號(〝–〞)用以描述普通的LOM資料型式及LOM資料元件，指出當LOM名稱因爲參照一個資料型式以代替LOM資料元件，而沒有XML名稱存在的時候。

(3) 子元件：LOM資料元件中子元件的一列表，一個〝None〞之進入代表LOM資料元件沒有任何子元件。如果顯示此項，則此子元件之列表應包含在該子元件之關聯聚合元件內，以確保LOMv1.0基礎架構部分之聚合關聯性能被強制遵守。各子元件之間的出現次序不重要。

(4) 最小值：如果LOM資料元件不是最上層之LOM元件，則LOM資料元件最小次數值的需求會出現在LOM資料元件的聚合元件情境中之LOM XML實例中。對於LOM元件而言，LOM資料元件最小次數值的需求可出現在LOM XML實例當中。

(5) 最大值：如果LOM資料元件並不是最上層之LOM元件，則LOM資料元件最大次數值的需求會出現在LOM資料元件的聚合元件情境中之LOM XML實例中。對於LOM元件而言，LOM資料元件最大次數值的需求可出現在LOM XML實例當中。本標準使用無窮大記號(〝∞〞)指示該最大次數值爲未設限，括號內的值爲其SPM。

(6) 次序性：指示值的次序性是否有意義，本標準採用三種次序性的說明：〝有次序性〞、〝無次序性〞、以及〝未規定〞。〝有次序性〞表示值之間的次序是有意義的，〝無次序性〞表示值之間的次序是無意義的，若LOM資料元件包含一個或多個的0，則其次序性的觀念不具任何含義並顯示爲〝未規定〞。

備考：一個列表中排序好的值其次序性的意義是根據其實作所決定。

(7) LOM資料型式：LOM資料型式之定義包含以下幾種型式：語言字串(LangString)、日期時間(DateTime)、期間(Duration)、詞彙(Vocabulary)、字元串(CharacterString)以及未規定的。如果LOM資料元件是屬於聚合元件之一，則其無資料型式，因此描述爲〝未規定〞。若資料

型式爲詞彙(Vocabulary)，則其允許之值必須參考LOMv1.0基礎架構所表列之值。

每個表格所描述的聚合元件包含其直屬子元件。舉例來說，第5.4節中表1描述LOM資料元件及其包含的直屬子元件，如一般(General)與生命週期(Life Cycle)。然而此表並未包含一般(General)元件之子元件，如識別符(Identifier)與標題(Title)。若該子元件也是聚合元件之一，則將會有一個獨立的表格描述該子元件之直屬子元件。若IEEE 1484.12.1-2002標準需要提供更進一步的資訊以定義LOM XML實例中之子元件，則此資訊將顯示在聚合元件描述之小節中。

除了描述(Description)、教育(Educational)、識別符(Identifier) 和適用對象(Intended End User Role)子元件之次序性以外，任何在第5.4節與第5.5節中表格中以及IEEE 1484.12.1-2002標準中不一致之處之值應使用IEEE 1484.12.1-2002標準所定義之值。儘管〝最大值〞一欄可指示子元件在聚合元件中可能顯示無設限的最大次數值，LOM XML實例的應用程式應處理不少於子元件的SPM數目(參照第5.1.1節)。

This Standard uses tables to describe the requirements for a LOM XML instance. The tables express requirements for each LOM data element in the LOMv1.0 base schema. These requirements are as follows:

— *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 5.5, 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, which 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 requirement 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 requirement 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 requirement 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 requirement on the maximum number of times the LOM element may appear in a LOM XML instance. An infinity symbol (“∞”) 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 “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 defined as 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.

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 Std 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 Std 1484.12.1-2002, the values from IEEE Std 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

表1描述LOM元件及其直屬子元件。

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

表1- LOM元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
LOM	lom	一般 生命週期 詮釋-詮釋資料 技術 教育 權利 關聯性 註解 分類	1	1	未規定	未規定
一般	general	參照第 5.4.1 節	0	1	未規定	未規定
生命週期	lifecycle	參照第 5.4.2 節	0	1	未規定	未規定

詮釋-詮釋資料	metaMetadata	參照第 5.4.3 節	0	1	未規定	未規定
技術	technical	參照第 5.4.4 節	0	1	未規定	未規定
教育	educational	參照第 5.4.5 節	0	∞ (100)	無次序性	未規定
權利	rights	參照第 5.4.6 節	0	1	未規定	未規定
關聯性	relation	參照第 5.4.7 節	0	∞ (100)	無次序性	未規定
註解	annotation	參照第 5.4.8 節	0	∞ (30)	無次序性	未規定
分類	classification	參照第 5.4.9 節	0	∞ (40)	無次序性	未規定

Table 1—The LOM element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
LOM	lom	General Life Cycle Meta-Metadata Technical Educational Rights Relation Annotation Classification	1	1	Unspecified	Unspecified
General	general	See 5.4.1	0	1	Unspecified	Unspecified
Life Cycle	lifeCycle	See 5.4.2	0	1	Unspecified	Unspecified
Meta-Metadata	metaMetadata	See 5.4.3	0	1	Unspecified	Unspecified
Technical	technical	See 5.4.4	0	1	Unspecified	Unspecified
Educational	educational	See 5.4.5	0	∞ (100)	Unordered	Unspecified
Rights	rights	See 5.4.6	0	1	Unspecified	Unspecified
Relation	relation	See 5.4.7	0	∞ (100)	Unordered	Unspecified
Annotation	annotation	See 5.4.8	0	∞ (30)	Unordered	Unspecified
Classification	classification	See 5.4.9	0	∞ (40)	Unordered	Unspecified

名稱空間宣告

LOM XML實例應包含一個名稱空間宣告用以宣告LOM元件及其組件之LOM名稱空間。如同第5.2節中的定義，本LOM名稱空間應爲〝http://ltsc.ieee.org/xsd/LOM〞。本名稱空間宣告之使用係符合〝W3C Recommendation, Namespaces in XML1.1.〞，圖1及圖2各自爲其舉例說明。

圖1. 預設名稱空間宣告之範例

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

</lom>

備考：名稱空間宣告允許應用程式識別LOM XML實例，其包含本標準所描述之LOM資料元件。

圖2. 特定前綴名稱空間宣告之範例

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

. . .

</lom:lom>

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 is in conformance with "W3C Recommendation, Namespaces in XML1.1." Examples are shown in Figure 1 and Figure 2, respectively.

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

. . .

</lom>

Figure 1—An example default 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.

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

. . .

</lom:lom>

Figure 2—An example prefix-specific namespace declaration

5.4.1 一般 (General)

表2描述一般(General)元件及其直屬子元件。

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

表2-一般元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
一般	general	識別符 標題 語言 描述 關鍵字 涵蓋範圍	0	1	未規定	未規定

		結構 聚集層次				
識別符	identifier	參照第 5.4.1.1 節	0	∞ (10)	無次序性	未規定
標題	title	參照第 5.5.4 節	0	1	未規定	語言字串
語言	language	無	0	∞ (10)	無次序性	字元串
描述	description	參照第 5.5.4 節	0	∞ (10)	無次序性	語言字串
關鍵字	keyword	參照第 5.5.4 節	0	∞ (10)	無次序性	語言字串
涵蓋範圍	coverage	參照第 5.54 節	0	∞ (10)	無次序性	語言字串
結構	structure	參照第 5.5.5 節	0	1	未規定	詞彙
聚集層次	aggregationLevel	參照第 5.5.5 節	0	1	未規定	詞彙

Table 2—The General element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
General	general	Identifier Title Language Description Keyword Coverage Structure Aggregation Level	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.1.1	0	∞ (10)	Unordered	Unspecified
Title	title	See 5.5.4	0	1	Unspecified	LangString
Language	language	None	0	∞ (10)	Unordered	CharacterString
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString
Keyword	keyword	See 5.5.4	0	∞ (10)	Unordered	LangString
Coverage	coverage	See 5.5.4	0	∞ (10)	Unordered	LangString
Structure	structure	See 5.5.5	0	1	Unspecified	Vocabulary
Aggregation Level	aggregationLevel	See 5.5.5	0	1	Unspecified	Vocabulary

5.4.1.1 識別符(Identifier)

表3描述識別符(Identifier)元件及其直屬子元件。

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

表3-識別符元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
識別符	identifier	編目項目	0	∞ (10)	無次序性	未規定
編目	catalog	無	0	1	未規定	字元串
項目	entry	無	0	1	未規定	字元串

Table 3—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString
Entry	entry	None	0	1	Unspecified	CharacterString

5.4.1.2 語言(Language)

語言(Language)元件應爲一個字元串，其字元串之值應爲下列之一：

(1) 第5.5.4.1節所描述之格式與值

(2) none符記

The Language element shall be a character string. The value of the character string shall be one of the following:

— The format and values described in 5.5.4.1

— The token none

5.4.1.3 結構(Structure)

如果結構(Structure)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則結構(Structure)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) atomic

(2) collection

(3) networked

(4) hierarchical

(5) linear

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

— atomic

— collection
— networked
— hierarchical
— linear

5.4.1.4 聚集層次(Aggregation Level)

如果聚集層次(Aggregation Level)元件中之來源(Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則聚集層次(Aggregation Level)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) 1
(2) 2
(3) 3
(4) 4

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 生命週期(Life Cycle)

表4描述生命週期(Lifecycle)元件及其直屬子元件。

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

表4-生命週期元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
生命週期	lifecycle	版本 狀態 貢獻	0	1	未規定	未規定
版本	version	參照第 5.5.4 節	0	1	未規定	語言字串
狀態	status	參照第 5.5.5 節	0	1	未規定	詞彙
貢獻	contribute	參照第 5.4.2.2 節		∞ (30)	有次序性	未規定

Table 4—The Life Cycle element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Life Cycle	lifeCycle	Version Status Contribute	0	1	Unspecified	Unspecified
Version	version	See 5.5.4	0	1	Unspecified	LangString
Status	status	See 5.5.5	0	1	Unspecified	Vocabulary
Contribute	contribute	See 5.4.2.2		∞ (30)	Ordered	Unspecified

5.4.2.1 狀態(Status)

如果狀態(Status)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則狀態(Status)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) draft

(2) final

(3) revised

(4) unavailable

If the value space for the 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 貢獻(Contribute)

表5描述貢獻(Contribute)元件及其直屬子元件。

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

表5-貢獻元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
貢獻	contribute	角色 個體 日期	0	∞ (30)	有次序性	未規定
角色	role	參照第 5.5.5 節	0	1	未規定	詞彙
個體	entity	無	0	∞ (40)	有次序性	字元串

日期	date	參照第 5.5.5 節	0	1	未規定	日期時間

Table 5—The Contribute element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Contribute	contribute	Role Entity Date	0	∞ (30)	Ordered	Unspecified
Role	role	See 5.5.5	0	1	Unspecified	Vocabulary
Entity	entity	None	0	∞ (40)	Ordered	CharacterString
Date	date	See 5.5.5	0	1	Unspecified	DateTime

5.4.2.2.1 角色(Role)

如果角色(Role)元件中之來 源(Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則角色(Role)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) author
(2) publisher
(3) unknown
(4) initiator
(5) terminator
(6) validator
(7) editor
(8) graphical designer
(9) technical implementer
(10) content provider
(11) technical validator
(12) educational validator
(13) script writer
(14) instructional designer
(15) subject matter expert

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
— 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 個體(Entity)

個體(Entity)元件所擁有的值應爲文字字元串，且依據IETF RFC 2426:1998所定義的有效vCard正準詞彙表示。

備考：IETF RFC 2426:1998不依賴XML語法表達該有效vCard之內部結構。本標準出版時，無現存的W3C XML架構定義語言繫結之vCard規格存在。

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 詮釋-詮釋資料(Meta-Metadata)

表6描述詮釋-詮釋資料(Meta-Metadata)元件及其直屬子元件。

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

表6-詮釋-詮釋資料元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
詮釋-詮釋資料	metaMetadata	識別符 貢獻 詮釋資料架構 語言	0	1	未規定	未規定
識別符	identifier	參照第 5.4.3.1 節	0	∞ (10)	無次序性	未規定
貢獻	contribute	參照第 5.4.3.2 節	0	∞ (10)	有次序性	未規定
詮釋資料	metadataSchema	無	0	∞ (10)	無次	字元串

架構					序性	
語言	language	無	0	1	未規定	字元串

Table 6—The Meta-Metadata element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Meta-Metadata	metaMetadata	Identifier Contribute Metadata Schema Language	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.3.1	0	∞ (10)	Unordered	Unspecified
Contribute	contribute	See 5.4.3.2	0	∞ (10)	Ordered	Unspecified
Metadata Schema	metadataSchema	None	0	∞ (10)	Unordered	CharacterString
Language	language	None	0	1	Unspecified	CharacterString

5.4.3.1 識別符(Identifier)

表7描述識別符(Identifier)元件及其直屬子元件。

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

表7-識別符元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
識別符	identifier	編目 項目	0	∞ (10)	無次序性	未規定
編目	catalog	無	0	1	未規定	字元串
項目	entry	無	0	1	未規定	字元串

Table 7—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString
Entry	entry	None	0	1	Unspecified	CharacterString

5.4.3.2 貢獻(Contribute)

表8描述貢獻(Contribute)元件及其直屬子元件。

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

表8-貢獻元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
貢獻	contribute	角色 個體 日期	0	∞ (10)	有次序性	未規定
角色	role	參照第 5.5.5 節	0	1	未規定	詞彙
個體	entity	無	0	∞ (10)	有次序性	字元串
日期	date	參照第 5.5.5 節	0	1	未規定	日期時間

Table 8—The Contribute element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Contribute	contribute	Role Entity Date	0	∞ (10)	Ordered	Unspecified
Role	role	See 5.5.5	0	1	Unspecified	Vocabulary
Entity	entity	None	0	∞ (10)	Ordered	CharacterString
Date	date	See 5.5.5	0	1	Unspecified	DateTime

5.4.3.2.1 角色(Role)

如果角色(Role)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則角色(Role)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) creator

(2) validator

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:

— creator

— validator

5.4.3.2.2 個體(Entity)

個體(Entity)元件所擁有的值應爲文字字元串，且依據IETF RFC 2426:1998所定義的有效vCard正準詞彙表示。

備考：IETF RFC 2426:1998不依賴XML語法表達該有效vCard之內部結構。本標準出版時，無現存的W3C XML架構定義語言繫結之vCard規格存在。

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 詮釋資料架構(Metadata Schema)

對於完全符合本標準之LOM XML實例，若詮釋資料架構(Metadata Schema)元件(metadataSchema)呈現於LOM XML實例中，則此元件應包含LOMv1.0值(即<metadataSchema>LOMv1.0</metadataSchema>)。對於符合但非完全符合本標準之LOM XML實例(即包含延伸資料模型元件)，若詮釋資料架構(Metadata Schema)元件(metadataSchema) 呈現於LOM XML實例中，則此元件應包含LOMv1.0值，且額外metadataSchema之出現次數將列出其他所有的詮釋資料架構或創造LOM XML實例所使用之可信賴的規格。

For those LOM XML instances that are *strictly conforming* to this Standard, 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 instances that are *conforming* but not *strictly conforming* (i.e., contain extended data model elements), 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 語言(Language)

語言(Language)元件應爲字元串。此字元串之值應符合第5.5.4.1節中描述之格式與值域。

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

5.4.4 技術(Technical)

表9描述技術(Technical)元件及其直屬子元件。
Table 9 describes the Technical element and its direct-descendant subelements.

表9-技術元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM資料型式

技術	technical	格式 檔案大小 位置 需求 安裝說明 其他平台需求 期間	0	1	未規定	未規定
格式	format	無	0	∞ (40)	無次序性	字元串
檔案大小	size	無	0	1	未規定	字元串
位置	location	無	0	∞ (10)	有次序性	字元串
需求	requirement	參照第 5.4.4.3 節	0	∞ (40)	無次序性	未規定
安裝說明	installationRemarks	參照第 5.5.4 節	0	1	未規定	語言字串
其他平台需求	otherPlatformRequirements	參照第 5.5.4 節	0	1	未規定	語言字串
期間	duration	參照第 5.5.3 節	0	1	未規定	期間

Table 9—The Technical element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Technical	technical	Format Size Location Requirement Installation Remarks Other Platform Requirements Duration	0	1	Unspecified	Unspecified
Format	format	None	0	∞ (40)	Unordered	CharacterString
Size	size	None	0	1	Unspecified	CharacterString
Location	location	None	0	∞ (10)	Ordered	CharacterString
Requirement	requirement	See 5.4.4.3	0	∞ (40)	Unordered	Unspecified
Installation Remarks	installationRemarks	See 5.5.4	0	1	Unspecified	LangString
Other Platform Requirements	otherPlatformRequirements	See 5.5.4	0	1	Unspecified	LangString
Duration	duration	See 5.5.3	0	1	Unspecified	Duration

5.4.4.1 格式(Format)

格式(Format)元件所擁有之值應爲字元串且該值應爲下述所列之一：

(1) 來自RFC 2048中MIME(Multipurpose Internet Mail Extension)型態值之標準詞彙表示式的字母。

(2) 非數字之符記。

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

— The token non-digital

5.4.4.2 檔案大小(Size)

檔案大小(Size)元件所擁有之值應爲0-9數字的結合。此LOM XML實例中之檔案大小(Size)值應爲定義於XML架構衍生資料型式(nonNegativeInteger)的有效非負數的整數(參照XML架構，第2部)。

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 需求(Requirement)

表10描述需求(Requirement)元件及其直屬子元件。

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

表10-需求元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
需求	requirement	或組成	0	∞ (40)	無次序性	未規定
或組成	orComposite	參照第 5.4.4.3.1 節	0	∞ (40)	無次序性	未規定

Table 10—The Requirement element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Requirement	requirement	OrComposite	0	∞ (40)	Unordered	Unspecified
OrComposite	orComposite	See 5.4.4.3.1	0	∞ (40)	Unordered	Unspecified

5.4.4.3.1 或組成(OrComposite)

表11描述或組成(OrComposite)元件及其直屬子元件。

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

表11-或組成元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
或組成	orComposite	型式 名稱 最低版本 最高版本	0	∞ (40)	無次序性	未規定
型式	type	參照第 5.5.5 節	0	1	未規定	詞彙
名稱	name	參照第 5.5.5 節	0	1	未規定	詞彙
最低版本	minimumVersion	無	0	1	未規定	字元串
最高版本	maximumVersion	無	0	1	未規定	字元串

Table 11—The OrComposite element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
OrComposite	orComposite	Type Name Minimum Version Maximum Version	0	∞ (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 型式(Type)

型式(Type)與名稱(Name)元件應成對出現。若其中一個元件出現在LOM XML實例中，則另一元件亦應出現。若型式(Type)元件中之來源(Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則型式(Type)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) operating system

(2) browser

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 名稱(Name)

型式(Type)與名稱(Name)元件應成對出現。若其中一個元件出現在LOM XML實例中，則另一元件亦應出現。若名稱(Name)元件中之來源(Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，且型式(Type)元件中值(Value)子元件之值爲符記作業系統，則名稱(Name)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) pc-dos

(2) ms-windows

(3) macos

(4) unix

(5) multi-os

(6) none

若名稱(Name)元件中之來源(Source)子元件的值域是LOMv1.0基礎架構(即

<source>LOMv1.0</source>)，且型式(Type)元件中之值(Value)子元件的值爲符記瀏覽器，則名稱(Name)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) any
(2) netscape communicator
(3) ms-internet explorer
(4) opera
(5) amaya

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 Type element is the token browser, then the valid values for the Value subelement of the Name element shall come from th following list of tokens:
— any
— netscape communicator
— ms-internet explorer
— opera
— amaya

5.4.5 教育(Educational)

表12描述教育(Educational)元件及其直屬子元件。

Table 12 describes the Educational element and its direct-descendant subelements.

表12-教育元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
教育	educational	互動型式	0	∞	無次	未規定

		學習資源型式 互動程度 語意密度 適用對象 情境 適用年齡範圍 困難度 基本學習時數 描述 語言		(100)	序性	
互動型式	interactivityType	參照第 5.5.5 節	0	1	未規定	詞彙
學習資源型式	learningResourceType	參照第 5.5.5 節	0	∞ (10)	有次序性	詞彙
互動程度	interactivityLevel	參照第 5.5.5 節	0	1	未規定	詞彙
語意密度	semanticDensity	參照第 5.5.5 節	0	1	未規定	詞彙
適用對象	intendedEndUserRole	參照第 5.5.5 節	0	∞ (10)	有次序性	詞彙
情境	context	參照第 5.5.5 節	0	∞ (10)	無次序性	詞彙
適用年齡範圍	typicalAgeRange	參照第 5.5.4 節	0	∞ (5)	無次序性	語言字串
困難度	difficulty	參照第 5.5.5 節	0	1	未規定	詞彙
基本學習時數	typicalLearningTime	參照第 5.5.3 節	0	1	未規定	期間
描述	description	參照第 5.5.4 節	0	∞ (10)	無次序性	語言字串

語言	language	無	0	∞ (10)	無次序性	字元串

Table 12—The Educational element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Educational	educational	Interactivity Type Learning Resource Type Interactivity Level Semantic Density Intended End User Role Context Typical Age Range Difficulty Typical Learning Time Description Language	0	∞ (100)	Unordered	Unspecified
Interactivity Type	interactivityType	See 5.5.5	0	1	Unspecified	Vocabulary
Learning Resource Type	learningResourceType	See 5.5.5	0	∞ (10)	Ordered	Vocabulary
Interactivity Level	interactivityLevel	See 5.5.5	0	1	Unspecified	Vocabulary
Semantic Density	semanticDensity	See 5.5.5	0	1	Unspecified	Vocabulary
Intended End User Role	intendedEndUserRole	See 5.5.5	0	∞ (10)	Ordered	Vocabulary
Context	context	See 5.5.5	0	∞ (10)	Unordered	Vocabulary
Typical Age Range	typicalAgeRange	See 5.5.4	0	∞ (5)	Unordered	LangString
Difficulty	difficulty	See 5.5.5	0	1	Unspecified	Vocabulary
Typical Learning Time	typicalLearningTime	See 5.5.3	0	1	Unspecified	Duration
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString
Language	language	None	0	∞ (10)	Unordered	CharacterString

5.4.5.1 互動型式(Interactivity Type)

若互動型式(Interactivity Type)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則互動型式(Interactivity Type)元件中之值(Value)子元件的有效值應來自以下之符記：

(1)active

(2)expositive

(3)mixed

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 學習資源型式(Learning Resource Type)

若學習資源型式(Learning Resource Type)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則學習資源型式(Learning Resource Type)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) exercise
(2) simulation
(3) questionnaire
(4) diagram
(5) figure
(6) graph
(7) index
(8) slide
(9) table
(10) narrative text
(11) exam
(12) experiment
(13) problem statement
(14) self assessment
(15) lecture

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
— problem statement
— self assessment
— lecture

5.4.5.3 互動程度(Interactivity Level)

若互動程度(Interactivity Level)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則互動程度(Interactivity Level)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) very low
(2) low
(3) medium
(4) high
(5) very high

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 語意密度(Semantic Density)

若語意密度(Semantic Density)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則語意密度(Semantic Density)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) very low
(2) low
(3) medium
(4) high
(5) very high

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:

(1) very low
(2) low
(3) medium
(4) high
(5) very high

5.4.5.5 適用對象(Intended End User Role)

若適用對象(Intended End User Role)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則適用對象(Intended End User Role)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) teacher

(2) author

(3) learner

(4) manager

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

5.4.5.6 情境(Context)

若情境(Context)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則情境(Context)元件中之值(Value)子元件的有效值應來自以下之符記：

(1)school

(2)higher education

(3)training

(4)other

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 困難度(Difficulty)

若困難度(Difficulty)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則困難度(Difficulty)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) very easy

(2) easy

(3) medium

(4) difficult

(5) very difficult

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 語言(Language)

此語言(Language)元件應爲字元串，此字元串之值應符合第5.5.4.1節中描述之格式與值域。

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

5.4.6 權利(Rights)

表13描述權利(Rights)元件及其直屬子元件。

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

表13-權利元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
權利	rights	價格 著作權及其他的限制 描述	0	1	未規定	未規定
價格	cost	參照第 5.5.5 節	0	1	未規定	詞彙
著作權及其他的限制	copyrightAndOtherRestrictions	參照第 5.5.5 節	0	1	未規定	詞彙
描述	description	參照第 5.5.4 節	0	1	未規定	語言字串

Table 13—The Rights element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Rights	rights	Cost Copyright And Other Requirements Description	0	1	Unspecified	Unspecified
Cost	cost	See 5.5.5	0	1	Unspecified	Vocabulary
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 價格(Cost)

若價格元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則價格元件中之值(Value)子元件的有效值應來自以下之符記：

(1) yes

(2) no

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 著作權及其他的限制(Copyright And Other Restrictions)

若著作權及其他的限制(Copyright And Other Restrictions)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則著作權及其他的限制(Copyright And Other Restrictions)元件中之值(Value)子元件的有效值應來自以下之符記：

(1)yes

(2)no

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 of tokens:

— yes

— no

5.4.7 關聯性(Relation)

表14描述關聯性(Relation)元件及其直屬子元件。

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

表14-關聯性元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
關聯性	relation	種類 資源	0	∞ (100)	無次序性	未規定
種類	kind	參照第 5.5.5 節	0	1	未規定	詞彙
資源	resource	參照第 5.4.7.2 節	0	1	未規定	未規定

Table 14—The Relation element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Relation	relation	Kind Resource	0	∞ (100)	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 種類(Kind)

若種類(Kind)元件中之來源(Source)子元件的值域是LOMv1.0基礎架構(即<source>LOMv1.0</source>)，則種類(Kind)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) ispartof
(2) haspart
(3) isversionof
(4) hasversion
(5) isformatof
(6) hasformat
(7) references
(8) isreferencedby
(9) isbasedon
(10) isbasisfor
(11) requires
(12) isrequiredby

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:

— ispartof

— haspart
— isversionof
— hasversion
— isformatof
— hasformat
— references
— isreferencedby
— isbasedon
— isbasisfor
— requires
— isrequiredby

5.4.7.2 資源(Resource)

表15描述資源(Resource)元件及其直屬子元件。

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

表15-資源元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
資源	resource	識別符 描述	0	1	未規定	未規定
識別符	identifier	參照第 5.4.7.2.1 節	0	∞ (10)	無次序性	未規定
描述	description	參照第 5.5.4 節	0	∞ (10)	無次序性	語言字串

Table 15—The Resource element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Resource	resource	Identifier Description	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.7.2.1	0	∞ (10)	Unordered	Unspecified
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString

5.4.7.2.1 識別符(Identifier)

表16描述識別符(Identifier)元件及其直屬子元件。

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

表16-識別符元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
識別符	identifier	編目項目	0	∞ (10)	無次序性	未規定
編目	catalog	無	0	1	未規定	字元串
項目	entry	無	0	1	未規定	字元串

Table 16—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString
Entry	entry	None	0	1	Unspecified	CharacterString

5.4.8 註解(Annotation)

表17描述註解(Annotation)元件及其直屬子元件。

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

表17-註解元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
註解	annotation	個體 日期 描述	0	∞ (30)	無次序性	未規定
個體	entity	無	0	1	未規定	字元串
日期	date	參照第 5.5.2 節	0	1	未規定	日期時間
描述	description	參照第 5.5.4 節	0	1	未規定	語言字串

Table 17—The Annotation element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Annotation	annotation	Entry Date Description	0	∞ (30)	Unordered	Unspecified
Entity	entity	None	0	1	Unspecified	CharacterString
Date	date	See 5.5.2	0	1	Unspecified	DateTime
Description	description	See 5.5.4	0	1	Unspecified	LangString

5.4.8.1 個體(Entity)

個體(Entity)元件所擁有的值應爲文字字元串，且依據IETF RFC 2426:1998所定義的有效vCard正準詞彙表示。

備考：IETF RFC 2426:1998不依賴XML語法表達該有效vCard之內部結構。本標準出版時，無現存的W3C XML架構定義語言繫結之vCard規格存在。

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.

5.4.9 分類(Classification)

表18描述分類(Classification)元件及其直屬子元件。

Table 18 describes the Classification element and its direct-descendant subelements.

表18-分類元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
分類	classification	目的 類路徑 描述 關鍵字	0	∞ (40)	無次序性	未規定
目的	purpose	無	0	1	未規定	詞彙
類路徑	taxonPath	參照第 5.4.9.2 節	0	∞ (15)	無次序性	未規定

描述	description	參照第 5.5.4 節	0	1	未規定	語言字串
關鍵字	keyword	參照第 5.5.4 節	0	∞ (40)	有次序性	語言字串

Table 18—The Classification element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Classification	classification	Purpose Taxon Path Description Keyword	0	∞ (40)	Unordered	Unspecified
Purpose	purpose	None	0	1	Unspecified	Vocabulary
Taxon Path	taxonPath	See 5.4.9.2	0	∞ (15)	Unordered	Unspecified
Description	description	See 5.5.4	0	1	Unspecified	LangString
Keyword	keyword	See 5.5.4	0	∞ (40)	Ordered	LangString

5.4.9.1 目的(Purpose)

若目的(Purpose)元件中之來 源 (Source)子元件的值域是LOMv1.0基礎架構(即 <source>LOMv1.0</source>)，則目的(Purpose)元件中之值(Value)子元件的有效值應來自以下之符記：

(1) discipline

(2) idea

(3) prerequisite

(4) educational objective

(5) accessibility restrictions

(6) educational level

(7) skill level

(8) security level

(9) competency

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 類路徑(Taxon Path)

表19描述類路徑(Taxon Path)元件及其直屬子元件。

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

表19-類路徑元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
類路徑	taxonPath	來源 類	0	∞ (15)	無次序性	未規定
來源	source	參照第 5.5.4 節	0	1	未規定	語言字串
類	taxon	參照第 5.4.9.2.1 節	0	∞ (15)	有次序性	未規定

Table 19—The Taxon Path element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Taxon Path	taxonPath	Source Taxon	0	∞ (15)	Unordered	Unspecified
Source	source	See 5.5.4	0	1	Unspecified	LangString
Taxon	taxon	See 5.4.9.2.1	0	∞ (15)	Ordered	Unspecified

5.4.9.2.1 類(Taxon)

表20描述類(Taxon)元件及其直屬子元件。

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

表20-類元件

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
類	taxon	識別符 項目	0	∞ (15)	有次序性	未規定
識別符	id	無	0	1	未規定	字元串
項目	entry	參照第 5.5.4 節	0	1	未規定	語言字串

Table 20—The Taxon element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Taxon	taxon	Id Entity	0	∞ (15)	Ordered	Unspecified
Id	id	None	0	1	Unspecified	CharacterString
Entry	entry	See 5.5.4	0	1	Unspecified	LangString

5.5 共通資料型式與元件

LOM XML架構繫結中所使用之資料型式定義於第5.5.1至第5.5.5節中。

Data types that are used in the LOM XML Schema binding are defined in 5.5.1 – 5.5.5.

5.5.1 字元串(CharacterString)

本標準所定義之元件其LOM資料型式爲字元串(CharacterString)者，W3C XML架構定義語言繫結應擁有字串之XML架構原始資料型式。此字串之資料型式支援完整ISO/IEC 10646-1標準，確保其支援包含多位元語言在內之多國語言。

備考：XML架構定義語言之原始資料型式字串定義於〝XML架構，第2部〞。

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 日期時間(DateTime)

日期時間(DateTime)資料型式以一組XML元件呈現。擁有日期時間(DateTime)資料型式之LOM資料元件可包含下列之子元件：

(1) 日期時間(DateTime)

(2) 描 述 (Description)

若出現上述之子元件，則應包含於被定義之日期時間(DateTime)資料型式之子元件內，該子元件之次序性不重要。表21描述日期時間(DateTime)資料型式及其直屬子元件。

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, these subelements 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.

表21-日期時間資料型式

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
日期時間	—	日期時間描述	0	1	未規定	未規定
日期時間	dateTime	無	0	1	未規定	字元串
描述	description	參照第 5.5.4 節	0	1	未規定	語言字串

備考：日期時間(DateTime)之資料型式係由日期時間(DateTime)及描述(Description)兩個元件所組成，亦爲所有定義爲日期時間(DateTime)型式之聚合元件的直屬元件。舉例來說，日期時間(DateTime)及描述(Description)爲聚合元件日期(Date)之直屬元件。(參照第 5.4.2.2 節)

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).

Table 21—The DateTime data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
DateTime	–	DateTime Description	0	1	Unspecified	Unspecified
DateTime	dateTime	None	0	1	Unspecified	CharacterString
Description	description	See 5.5.4	0	1	Unspecified	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 日期時間(DateTime)

日期時間(DateTime)元件應爲根據下列限制定義之一種型樣。

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

其中

YYYY是四位數之年份(該值大於等於0001)，

MM是兩位數之月份(01到12，01代表一月，以此類推)，

DD是兩位數之日期(01到31，該值根據年份及月份而異)，

hh是兩位數之小時(00到23，不允許使用AM與PM)，

mm是兩位數之分鐘(00到59)，

ss是兩位數之秒鐘(00到59)，

s是一個或多個數字用以表示小數後之秒鐘，

TZD是時區指定器(“Z”表示協調標準時間(或世界標準時間)UTC或使用+hh, -hh, +hh:mm,或 -hh:mm，其中hh代表兩位數之小時，mm代表兩位數之分鐘)

若DateTime元件包含一個值，則至少應包含四位數之年份。若亦包含日期與時間，DateTime元件使用之字元字母爲“-”, “T”, “:”, 以及 “.”。

若有顯示時間部分，但未指定時區，則時區將自動使用UTC表示。

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

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

where

YYYY is the four-digit year (≥0001)

MM is the two-digit month (01 through 12, where 01 = January, etc.)

DD is the two-digit day of a month (01 through 31, depending on the values of month and year)

hh are the two digits of an hour (00 through 23) (AM and PM are NOT allowed)

mm are the wo digits of a minute (00 through 59)

ss are the two digits of a second (00 through 59)

s is one or more digits representing a decimal fraction of a second

TZD is the time zone designator (“Z” for coordinated universal time [UTC] or +hh, -hh, +hh:mm, or -hh:mm, where hh is two digits of an hour and mm is two digits of a 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.

備考：1.因爲限制取代了值域，W3C XML架構定義語言原始的資料型式dateTime不能被用以建立符合之LOM XML實例，因此，正常的表達方式爲呈現所有IEEE1484.12.1-2002標準中有效的日期與時間。

2.部分的日期只呈現日期於Common Era(CE)。部分的日期遵守制訂於1582年10月15的標準日期，在之前，則是使用Julian格式，獨立於特定場所。在BCE前的日期和其他應該用描 述(Description)元件(參照第5.5.2節)呈現的案例。

3.同層次的meta特性(“[“and”]”)指出選項的元件可出現0或1次於元件詞彙的日期時間(DateTime)元件表示方式。此meta特性並不會出現在結果中，只會連結到描述的(例：DD被相對應的兩位值日期所表示)值中。

4.值域以ISO 8601:2000[B3]爲基礎。

NOTE 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 Std 1484.12.1-2002.

NOTE 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 before 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).

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

NOTE 4—The value space is based on ISO 8601:2000 [B3].

5.5.3 期間(Duration)

期間(Duration)資料型式以一組XML元件呈現，擁有期間(Duration)資料型式之LOM資料元件可包含下列之子元件：

(1) 期間(Duration)

(2) 描述(Description)

若出現上述之子元件，則此子元件應被包含於定義的期間(Duration)資料型式之元件中，此子元件出現之次序性不重要。

表22描述期間(Duration)資料型式及其直屬子元件。

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

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

表22-期間資料型式

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
期間	—	期間描述	0	1	未規定	未規定
期間	duration	無	0	1	未規	字元串

					定	
描述	description	參照第 5.5.4 節	0	1	未規定	語言字串

備考：期間(Duration)之資料型式係由期間(Duration)及描述(Description)兩個元件所組成，亦爲所有定義爲期間(Duration)型式之聚合元件的直屬元件。舉例來說，期間(Duration)及描述(Description) 爲聚合元件期間(Duration)之直屬元件。(參照第 5.4.4 節)

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).

Table 22—The Duration data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Duration	–	Duration Description	0	1	Unspecified	Unspecified
Duration	duration	None	0	1	Unspecified	CharacterString
Description	description	See 5.5.4	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 期間(Duration)

期間(Duration)元件之值域應爲根據下列限制定義之一種型樣：

P[yY][mM][dD][T[hH][nM][s[.s]S]]

其中

y代表年份之數字(大於或等於0之整數)

m代表月份之數字(大於或等於0之整數，不限制，如，允許其值>12)

d代表日期之數字(大於或等於0之整數，不限制，如，允許其值>31)

h代表小時之數字(大於或等於0之整數，不限制，如，允許其值>23)

n代表分鐘之數字(大於或等於0之整數，不限制，如，允許其值>59)

s代表秒鐘或秒鐘的再劃分之數字(大於或等於0之整數，不限制，如，允許其值>59)

若以上相對應的值是非0值，則下列字元字母“P”、“Y”、“M”、“D”、“T”、“H”、“M”、和“S”應同時呈現之。

若期間(Duration)元件包含一個值，則“P”符號應被呈現。若年份、月份、日期、小時、分鐘或者秒鐘之值爲0，則該表達之字母(例如“M”)可省略，但至少需包含有一個指定字母且該值爲正數除“P”之外。若所有的時間值均爲0(小時、分鐘以及秒鐘)則指定字母“T”應省略。

本標準並不支援負數期間。

The value space for the Duration element is a pattern that shall be defined according to the following constraints:

P[yY][mM][dD][T[hH][nM][s[.s]S]]

where

y is the number of years (integer, ≥ 0)

m is the number of months (integer, ≥ 0, not restricted, e.g., >12 is acceptable)

d is the number of days (integer, ≥ 0, not restricted, e.g., >31 is acceptable)

h is the number of hours (integer, ≥ 0, not restricted, e.g., >23 is acceptable)

n is the number of minutes (integer, ≥ 0, not restricted, e.g., >59 is acceptable)

s is the number of seconds or fraction of seconds (integer, ≥0, not restricted, e.g., >59 is acceptable)

The character literal designators “P”, “Y”, “M”, “D”, “T”, “H”, “M”, and “S” 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 integer value shall be present in addition to the designator “P”. The designator “T” shall be omitted if all of the time (hours, minutes, and seconds) is zero.

Negative durations are not supported.

備考：1.因爲限制取代了值域，W3C XML架構定義語言之原始資料型式dateTime不能被用以建立符合的LOM XML實例。正常的表達方式爲呈現所有在IEEE1484.12.1-2002中有效的日期/時間。

2.值被指定在於Gregorian calendar中。

3.期間之安排可能爲不確定的(例：一個月可能有28、29、30或31日)。

4.同層次的meta特性(“[“and”]”)指出選項的元件可出現0或1次於元件詞彙的期間(Duration)元件表示方式。此meta特性並不會出現在結果中，只會連結到描述的(例：dD是被在日期中遵守D的相對應的值所取代)值中。

5.值域以ISO8601:2000[B3]爲基礎的。

NOTE 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 Std 1484.12.1-2002.

NOTE 2—The value is designated in the Gregorian calendar.

NOTE 3—The ordering of durations may be indeterminate (e.g., 1 month may be 28, 29, 30, or 31 days).

NOTE 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 designator “D”).

NOTE 5—The value space is based on ISO 8601:2000 [B3].

5.5.3.2 描述(Description)

描述(Description)資料型式以語言字串(LangString)資料型式(參照第5.5.4節)呈現。

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

5.5.4 語言字串(LangString)

語言字串(LangString)資料型式以一組XML元件呈現，擁有語言字串(LangString)資料型式之LOM資料元件可包含下列之子元件：

(1)字串(String)

(2)語言(Language)

若出現上述之子元件，則此子元件應包含於被定義之語言字串(LangString)資料型式之元件中，此子元件出現之次序性不重要。在LOM XML實例中，LOM之語言資料元件應依照語言字串(LangString)型式之LOM資料元件屬性來規定。

表23描述語言字串(LangString)資料型式及其直屬子元件。

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, these subelements 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.

表23-語言字串資料型式

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
語言字串	—	字串 語言	0	(10)	無次序性	未規定
字串	string	無	0	1	未規定	字元串
語言	language	無	0	1	未規定	字元串
備考：語言字串(LangString)之資料型式由字串(String)及語言(Language)兩個元件所組成，亦爲所有定義爲語言字串(LangString)型式之聚合元件的直屬元件。舉例來說，字串(String)及語言(Language)爲聚合元件標題(Title)之直屬元件。(參照第5.4.1節)						

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)

Table 23—The LangString data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
LangString	–	String Language	0	(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 語言(Language)

語言(Language)元件應爲多種、選項的、使用連字號-前綴之需求語言編碼所構成之字元串，以下之限制適用於語言編碼中字元串的部分：

(1) 雙字母之編碼定義於ISO 639-1標準。

(2) 三字母之編碼定義於ISO 639-2標準。

(3) 單一字母“i”被保留爲IANA標準中註冊之前綴使用。

(4) 單一字母“x”被保留爲自定義之前綴使用。

以下之限制適用於第一子編碼中字元串的部分：

(1) 雙字母之子編碼屬於ISO 3166-1 alpha-2 country codes。

(2) 三至八字元長度之子編碼則規範於IANA標準中。

未規定額外子編碼之限制。

字元串所擁有之值應採用XML架構衍生資料型式語言定義之有效語言編碼(參照XML架構，第2部)。ISO 639-2 規定兩項編碼集合，其一爲編目應用(ISO 639-2/B)，另一個爲術語應用(ISO 639-2/T)，兩者皆可使用。

備考：1.語言編碼一般利用小寫字母表示，其子編碼(若有的話)則利用大寫字母表示，然而，對此值而言，大小寫英文子母都視爲相同(case insensitive)。

2.XML架構衍生資料型式語言不強制遵守所有在此語言碼中之限制。

範例

“en-GB”

“de”

“fr-CA”

“it”
“i-bnn” (IANA Bunun)

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:
— Two-letter codes are defined in ISO 639-1.
— Three-letter codes are defined in ISO 639-2.
— The one-letter code “i” is reserved and uses as a prefix for registrations defined by the IANA.
— The one-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:
— Two-letter subcodes are ISO 3166-1 alpha-2 country codes.
— Subcodes from three to eight characters in length 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.
NOTE 1—The language code is normally given in lower case and the subcodes (if any) in upper case. However, the values are case insensitive.
NOTE 2—The XML Schema derived data type language does not enforce all constraints on this language code.
Examples
“en-GB”
“de”
“fr-CA”
“it”
“i-bnn” (IANA Bunun)

5.5.5 詞彙(Vocabulary)

詞彙(Vocabulary)資料型式以一組XML元件呈現，詞彙(Vocabulary)資料型式之LOM資料元件可能包含下列之子元件：

(1) 來源(Source)

(2) 值(Value)

如果出現上述之子元件，則此子元件應被包含於定義於詞彙(Vocabulary)資料型式之元件中，此子元件之出現次序性不重要。

表24描述詞彙(Vocabulary)資料型式及其直屬子元件。

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, these subelements 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 24 describes the Vocabulary data type and its direct-descendant subelements

表24-詞彙資料型式

LOM 資料元件	XML 名稱	子元件	最小數量	最大數量	次序性	LOM 資料型式
詞彙	—	來源 值	未規定	未規定	未規定	未規定
來源	source	無	0	1	未規定	字元串
值	value	無	0	1	未規定	字元串

備考：詞彙(Vocabulary)資料型式係由來源(Source)及值(Value)兩個元件所組成，亦爲所有定義爲詞彙(Vocabulary)型式之聚合元件的直屬元件。舉例來說，來源(Source)及值(Value)爲聚合元件聚合等級(Aggregation Level)之直屬元件。(參照第 5.4.1.4 節)

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).

Table 24—The Vocabulary data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Vocabulary	–	Source 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 值(Value)

字元串所擁有之值應爲XML架構衍生資料型式符記(參照XML架構，第2部)定義之有效的符記。屬於詞彙(Vocabulary)型式之LOM資料元件定義符記之有效值。

備考：XML架構衍生資料型式符記支援ISO/IEC10646-1的所有定義。此符記資料型式確保其支援包含多位元語言在內之多國語言。多位元語言不被LOM 1.0基礎架構之詞彙使用，但可能用於不在LOM 1.0中之詞彙來源中。

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.

NOTE—The XML Schema derived data type token supports the repertoire of ISO/IEC 10646-1. The token data type ensures the support of multiple languages including multibyte languages. 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.

附錄A

(參考)

參考文獻

[B1] IEEE 100TM, The Authoritative Dictionary of IEEE Standards Terms.

[B2] IETF RFC 2425:1998, MIME Content-Type for Directory Information.

[B3] ISO 8601:2000, Data Elements and Interchange Formats—Information Interchange—Representation o f Dates and Times.

[B4] ISO/IEC 11404:1996, Information Technology—Programming Languages, Their Environments and System Software Interfaces—Language-Independent Datatypes.

[B5] W3C Recommendation (4 February 2004), Extensible Markup Language (XML) 1.1.

[B6] W3C Recommendation (4 February 2004), XML Information Set, Second Edition

Annex A

(informative)

Bibliography

[B1] IEEE 100TM, The Authoritative Dictionary of IEEE Standards Terms.

[B2] IETF RFC 2425:1998, MIME Content-Type for Directory Information.

[B3] ISO 8601:2000, Data Elements and Interchange Formats—Information Interchange—Representation o f Dates and Times.

[B4] ISO/IEC 11404:1996, Information Technology—Programming Languages, Their Environments and System Software Interfaces—Language-Independent Datatypes.

[B5] W3C Recommendation (4 February 2004), Extensible Markup Language (XML) 1.1.

[B6] W3C Recommendation (4 February 2004), XML Information Set, Second Edition.

附錄B

(參考)

網際網路上取得之XSD檔案

網際網路上可取得XSD檔案和範例，然而並未定義爲此標準中的一部份，或因爲任何特定目標被IEEE認證，必須自己承擔使用上的風險。

XSD檔案可在下列網址中取得：

http://standards.ieee.org/reading/ieee/downloads/LOM/lomv1.0/

備考：本標準不需要使LOM XML實例有效之XSD用法。

Annex B

(informative)

Internet availability of XSD files

XSD files are available as examples on the World Wide Web; however, they are not to be construed as a part of this Standard, nor endorsed by the IEEE for any specific purpose. USE AT YOUR OWN RISK.

The XSD files are available at the following URL:

http://standards.ieee.org/reading/ieee/downloads/LOM/lomv1.0/

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

附錄C

(參考)

XSD檔案描述

LOM 1.0基礎架構描述標準資料項之結構收錄，包含其資料型式、多樣性以及容器/組件間的關聯性，但沒有提供符合LOM 1.0基礎架構編碼LOM資料的語法。

本標準中描述的LOM XML繫結定義了編碼LOM資料之XML語法。繫結即是規則的收錄，用以描述如何在XML語法中表達LOM資料。此繫結的規則將詳細定義於第5節中。

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 it 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.

當建立XSD時，實作者能決定何時強制遵守LOM 1.0基礎架構之限制。此限制要能被定義於XSD中，以架構處理器履行限制之確認或藉由處理PSVI以履行該確認。第5節中定義建立有效LOM實例之準則；XSD實作者必須決定如何根據限制以發展XSD架構檔案。無論如何實作，符合與完全符合之LOM實例必須符合第5節中定義之規則。

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 the PSVI. 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 *conforming* or *strictly conforming* LOM instance has to conform to the rules defined in Clause 5.

參考W3C XML XSD使實作者能滿足許多在IEEE 1484.12.1-2002(參照附錄B)標準中的需求，和既有標準採用者的實行。XSD主要是用以支援W3C XML架構定義語言(參照XML架構,第1部與第2部)之XML處理工具。LOM 1.0基礎架構把需求建立在LOM XML實例上，有些無法用W3C XML架構定義語言表示。此需求定義於第5節和LOM 1.0基礎架構中。

主要目的爲提供替代的XSD，每個皆支援LOM XML架構繫結並確定選擇在XSD中用那個繫結限制描述。XSD中定義之限制應該藉由支援XML架構確認的工具強制遵守。沒有在XSD中表示之限制會藉由其他如處理PSVI的工具強制遵守。不論限制之強制遵守在那裡履行，符合或完全符合之LOM實例必須符合所有定義於第5節及LOM 1.0基礎架構中之限制。

Informative W3C XML XSDs are available for implementers to satisfy many requirements of IEEE Std 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 and in 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 and in the LOMv1.0 base schema.

一個預先定義之合成XSD(參照附錄C1.1)在呈現上可以多樣的型式出現，例如：特定的LOM資料元件排序、詞彙值之檢查以及延伸之使用。三個額外之預先定義的合成XSD是爲了強制遵守不同的確認限制(參照附錄C1.2至附錄C1.3)。四個合成XSD係由數個組件XSD集合而形成(參照附錄C.2)。應用程式開發者因爲在使用XSD上有其不同的需求目的，因此合成XSD必須是有選擇性的。提供替代XSD之方式闡明了不同採用社群之不同需求，進而在社群的處理過程與使用需求中產生符合及完全符合LOM XML實例之限制。

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 needs of the various adopting communities to produce *conforming* or *strictly conforming* LOM XML instances consistent with the communities' processing and usage requirements.

任何特定XSD之LOM XML架構繫結的部分，不論是存在或不存在之強制遵守，皆無法在考慮成爲符合或完全符合之LOM XML實作上，減少一個滿足所有LOM XML架構繫結限制之特定LOM XML實例。本標準中，存在之替代XSD只提供開發者在決定如何強制遵守LOM XML架構繫結限制之選擇。

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 LOM XML Schema binding constraints 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 合成XSD

lom.xsd合成XSD在其他的合成XSD(lomLoose.xsd、lomStrict.xsd以及lomCustom.xsd)被模組化之後，成爲一個模版。其中每一個XSD皆被附錄C.2中之組件XSD組織起來。藉由預設lom.xsd合成XSD以編碼W3C XML架構定義語言中之一組特定繫結限制，以致於LOM XML實例將完全符合IEEE1484.12.1-2002標準。

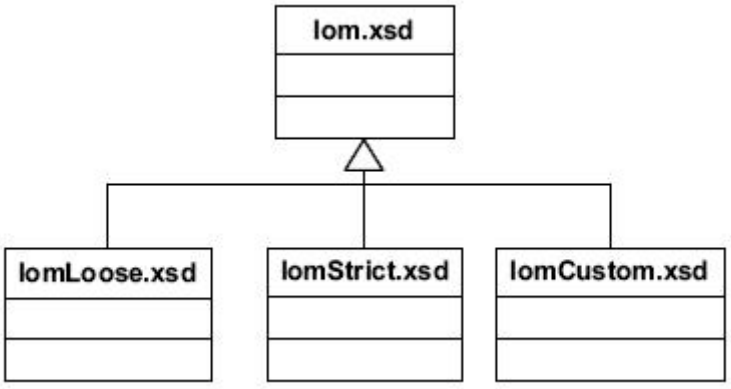
如圖C.1所示，lom.xsd合成XSD爲一抽象之合成XSD，由三個額外之合成XSD(lomLoose.xsd、lomStrict.xsd、lomCustom.xsd)所建立。此合成XSD模組了某些基於結合多個組件XSD選擇之確認方法。(參照C1.1至C1.3可得到更多此合成XSD之資訊)

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 Std 1484.12.1-2002.

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.)

圖C.1 合成XSD之關聯性

Figure C.1—The relationship of the composite XSDs



備考：在C1.1至C 1.3中之XSD名稱，lomLoose.xsd、lomStrict.xsd、及lomCustom.xsd，是參考由URL取出之檔案(參照附錄B)。

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).

C.1.1 lomStrict.xsd

lomStrict.xsd合成XSD編碼在W3C XML架構定義語言中之一組特定繫結限制，以致於LOM XML

實例將完全符合IEEE1484.12.1-2002標準。

lomStrict.xsd合成XSD：

(1)第5節中定義之限制詞彙(來源/值 成對方式表示)。

(2)表示在第5節中定義之唯一性限制。

(3)不允許LOM 1.0基礎架構之延伸(參照第5.1.3節)。

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 Std 1484.12.1-2002.

The lomStrict.xsd composite XSD

— Limits vocabularies (source/value pairs) to those defined in Clause 5

— Expresses the uniqueness constraints defined in Clause 5

— Does not permit extensions to the LOMv1.0 base schema (see 5.1.3)

C.1.2 lomCustom.xsd

lomCustom.xsd合成XSD編碼在W3C XML架構定義語言中之一組特定繫結限制，以致於LOM XML實例將符合IEEE1484.12.1-2002標準。lomCustom.xsd合成XSD：

(1) 除了第5節中的定義之外，還定義了一個機制描述詞彙(來源/值 成對方式表示)。XML架構處理器將檢查LOM XML實例中之詞彙值是否符合第5節中之定義或符合延伸詞彙中之定義。

(2) 表示第5節中定義之唯一性限制。

(3) 定義一機制描述LOM 1.0基礎架構(參照第5.1.3節)之延伸。延伸之部分定義於一組支援XSD中。XML架構處理器會檢查在LOM XML實例中之延伸是否符合定義於支援XSD中包含延伸之限制。

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 Std 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. An XML schema processor 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.

— 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. An XML schema processor 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 lomLoose.xsd

lomLoose.xsd合成XSD編碼在W3C XML架構定義語言中之一組特定繫結限制，以致於LOM XML實例可能無法符合或完全符合IEEE1484.12.1-2002標準。lomLoose.xsd合成XSD：

(1) 並不對於任何定義詞彙值提供確認(即定義於第5節中之值，或者是定義於延伸詞彙中之值)。XML架構處理器將證明詞彙值爲唯一之有效符記。

(2) 沒有表示第5節定義中之唯一性限制。

(3) 定義一機制描述LOM 1.0基礎架構之延伸(參照第5.1.3節)。延伸之部分定義於一組支援XSD中。XML架構處理器會檢查在LOM XML實例中之延伸是否符合定義於支援XSD中包含延伸之限制。

備考：缺少詞彙值之確認並不會免除特定LOM XML實例去滿足第5節所定義的詞彙需求。使用 vocab/loose.xsd 組件XSD之應用將由其他方法強制遵守詞彙之需求。

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 Std 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 or values defined in an extended vocabulary). An XML schema processor will verify that the vocabulary values are valid tokens only.

— Does not express the uniqueness constraints defined in Clause 5.

— 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. An XML schema processor 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. Applications that use the vocab/loose.xsd component XSD should enforce those vocabulary requirements by other means.

C.2 組件XSD

組件XSD呈現合成XSD之構件。下述組件XSD之使用由C.1中之合成XSD來定義：

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

(1) common/anyElement.xsd：定義基礎XML架構模型，群組元件及屬性用於延伸資料元件及XML屬性。

(2) common/dataTypes.xsd：定義第5節定義中資料型式之全域資料型式宣告。

(3) common/elementNames.xsd：定義第5節定義中每一個資料元件之全域元件宣告。此組件XSD是用以檢查資料元件之唯一性，是否在其聚合元件中的uniqueElementName屬性出現時宣告爲

唯一。此XML架構限制唯一性是爲了強制遵守唯一性之限制。

(4) common/elementTypes.xsd：定義第5節定義中資料元件之全域資料型式宣告，此組件XSD定義LOM資料元件間的聚合關聯性。此聚合關聯性強制遵守LOM 1.0基礎架構需求，元件可以在LOM XML實例中被表示成其所屬的聚合元件。

(5) common/rootElement.xsd：在LOM XML實例中，爲包含全部其他LOM資料元件定義此元件之名稱宣告。

(6) common/vocabTypes.xsd：定義第5節定義中詞彙資料型式中取得值之LOM資料元件之全域資料型式宣告。

(7) common/vocabValues.xsd：定義第5節定義中標準詞彙值宣告，此宣告用於連結vocab/custom.xsd與vocab/strict.xsd。

(8) extend/custom.xsd： 定義XML架構模型群組customElements與customAttributes以支援延伸資料元件和屬性之確認。群組於組件XSD中之模型參考定義於common/anyElement.xsd中之模型群組，此組件XSD將使用於若延伸部分在LOM XML實例中被支援時(參照第5.1.3節)。

備考：由extend/custom.xsd 組件XSD採用適當的PSVI處理，則使用延伸部分之LOM XML實例將符合但不完全符合本標準。

(9) extend/strict.xsd: 定義XML架構模型群組customElements與customAttributes。此模型群組被定義爲空的模型群組並且與其他的組件XSD被用以支援完全符合LOM XML實例之確認。此組件XSD將被使用於若延伸部分沒有在LOM XML實例中被支援時。

備考：由extend/strict.xsd 組件XSD採用適當的PSVI處理，而且，也因此不支援延伸部分，則LOM XML實例將完全符合本標準。

(10) unique/loose.xsd：爲LOM資料元件定義XML架構模型群組宣告，以支援LOM XML實例中的架構-基礎之唯一性限制確認，與每一個元件有關聯之一組確切屬性必須由LOM XML架構繫結所規定(即：必須避開當額外屬性強制遵守唯一性時)。此組件XSD被用以減輕強制遵守唯一性限制時，避免介紹XML屬性uniqueElementName(參照附錄E 3.3)。

備考：不強制遵守唯一性之限制並不會解除第5節定義中滿足唯一性限制之LOM XML實例，用於unique/loose.xsd 組件XSD中的應用程式必須以其他方法強制遵守唯一性限制。

(11) unique/strict.xsd：定義第5節中LOM資料元件之XML架構模型群組宣告，以支援LOM XML實例中的架構-基礎之唯一性限制確認，爲每一個至多出現1次之LOM資料元件介紹uniqueElementName屬性(參照E 3.3)。

備考：對於大多數的應用程式，利用unique/strict.xsd的組件XSD實施唯一性限制是合意的，雖然增加uniqueElementName屬性爲不受歡迎的，但不大可能會造成問題。

(12) vocab/custom.xsd：強制遵守的詞彙值只有規定在第5節中及延伸詞彙中，此組件XSD強制遵守堅持兼具標準的使用加上延伸之詞彙值，並檢查兩者的來源和值是來自定義於第5節中之符記集合或是來自於延伸之詞彙。

備考：在某種情況下，運用vocab/custom.xsd的組件XSD以嚴格實施定義於第5節中之詞彙值和延伸詞彙值是合意的，但有可能會把架構確認過程複雜化。

(13) vocab/loose.xsd：強制遵守詞彙來源和值爲字元串，此組件XSD放鬆了確認限制，允許來源與值兩者成爲任意字元串。運用vocab/loose.xsd之組件XSD的LOM XML實例可能是不符合的。需要對來源/值 成對方式表示確認的應用程式將必須處理PSVI。

備考：不強制遵守由vocab/loose.xsd之組件XSD的詞彙值確認，並不會解除在第5節中描述的滿足詞彙限制之特定LOM XML實例。運用vocab/loose.xsd之組件XSD的應用程式必須以其他方式強制遵守詞彙限制。

(14) vocab/strict.xsd：只有在第5節中所規定的詞彙值會被強制遵守，此組件XSD不允許從其他來源而來之延伸詞彙值。此組件XSD藉由檢查來源與值是否來自定義於第5節中之符記集合以支援完全符合詞彙值之確認。

— 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.

— common/elementNames.xsd: Defines global element declarations for each data element defined in Clause 5. 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. 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 in Clause 5.

— common/vocabValues.xsd: Defines the standard vocabulary value declarations defined in Clause 5. 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, by using the extend/custom.xsd component XSD. LOM XML instances that use extensions will conform but not strictly conform to this Standard.

— extend/strict.xsd: 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 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 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. 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 for LOM data elements defined in Clause 5 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 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 and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but it may complicate 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 nonconforming. Applications that require validation of the source/value pairs will have to process the PSVI. *Note:* The absence of the enforcement of 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. Applications that use the vocab/ loose.xsd component XSD have to enforce those vocabulary constraints by other means.

— vocab/strict.xsd: Enforces that vocabulary values are only those specified in Clause 5. This component XSD does not allow extended vocabulary 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.

附錄D

(參考)

延伸資料元件及屬性之賦能

LOM 1.0基礎架構預計介紹延伸部分之需求，不包括在本標準元件之收錄中(參照第5.1.3節)，此延伸部分必須遵守下述限制：

(1)LOM 1.0基礎架構之延伸部分保留LOM 1.0基礎架構中的值域及LOM資料元件之資料型式。

(2)LOM 1.0基礎架構中之延伸部分沒有爲聚合元件而定義資料型式或值域。

此需求意指若有聚合元件包括延伸部分，則延伸部分不會重新定義該聚合元件預期的意義(即：值域和資料型式)。爲符合此需求，參考的架構(參照附錄B)包含任一對聚合元件的內容模型中之元件(<xs:any namespace="##other"processContents="lax"/>)。

此processContents="lax"之宣告指示XML處理器嘗試確認元件之內容。若元件的架構資訊可被找到的情況下，此宣告允許除了LOM 1.0基礎架構名稱空間以外的元件被包含，以及內容被確認。若此情況不被允許，則此XML架構處理器將確認任一良好完整格式之LOM XML實例。

備考：1. 爲確保建立符合LOM XML之實例，發展延伸部分的組織也應提供對延伸定義確認限制之XSD。

2. 提供延伸屬性的組織可能會選擇爲此延伸定義名稱空間，若一個組織定義一個名稱空間，則在附錄B中參考之XSD就必須改變以支援此名稱空間。

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 e lements from the LOMv1.0 base schema.

— 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.

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

NOTE 2—Organizations providing extensions may choose to define namespaces for these extensions. If an organization defines a namespace, then the XSDs referenced in Annex B may have to be altered to support the namespace.

D.1 延伸資料元件之賦能

LOM 1.0基礎架構資料元件可能被延伸，此延伸是除了定義於第5.2節之名稱空間外，定義在名稱空間中新的XML元件。延伸之資料元件可能增加到定義成聚合元件之任一LOM資料元件中。爲允許延伸資料元件，extend/custom.xsd之組件XSD將被用於建構合成XSD。此組件XSD定義XML架構模型群組customElements，並且參考定義於common/anyElement.組件XSD中之customElements模型群組(參照附錄B中之lomCustom.xsd檔案)。合成XSD之此兩個組件XSD的內容物允許延伸資料元件之確認。

相關的組件XSD和其如何允許延伸資料元件如下述：

(1)common/anyElement.xsd：對用做延伸資料元件之元件定義XML基礎架構模型群組。

(2)common/dataTypes.xsd , common/elementTypes.xsd：對每一個定義於組件XSD之聚合complexType宣告，定義一個參考customElements宣告之XML架構模型群組。

(3)common/vocabTypes.xsd：對每一個定義於此組件XSD中之complexType詞彙宣告，定義一個參考customElements宣告之XML架構模型群組。

(4)extend/custom.xsd：定義XML架構模型群組customElements以支援延伸資料元件之確認。在組件XSD中之XML架構模型群組，參考定義於common/anyElement.xsd之組件XSD的模型群組。

備考：使用合成XSD extend/strict.xsd時，延伸部分可能無法使用。

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 may 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 referenced in 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 as follows:

— 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 these component XSDs, 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 延伸屬性之賦能

LOM 1.0基礎架構資料元件可能與除了定義於第5.2節外之名稱空間中的XML屬性一起被延伸，延伸之XML屬性可能被附加到任何LOM資料元件中。爲允許延伸之屬性，extend/custom.xsd之組件XSD可能被用以建構合成XSD。此組件XSD定義XML架構模型群組customAttributes。此模型群組定義除了第5.2節定義之名稱空間外，從名稱空間增加任一屬性之能力。合成XSD之此組件XSD內容物允許延伸屬性之確認。

有關此組件XSD及其如何允許延伸屬性如下述：

(1) common/anyElement.xsd：對用做延伸資料元件之元件定義基礎XML架構模型群組。

(2) common/dataTypes.xsd，common/elementTypes.xsd：對每一個在組件XSD中之聚合complexType宣告，定義一個參考customAttributes宣告之XML架構模型群組。

(3) common/vocabTypes.xsd：對每一個在此組件XSD中之詞彙和complexType宣告值，定義一個參考customAttributes之XML架構模型群組宣告的XML架構attributeGroup。

(4) extend/custom.xsd：定義支援延伸屬性之確認的XML架構模型群組customAttributes。

備考：使用合成XSD extend/strict.xsd時，延伸部分可能無法使用。

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 may 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 as follows:

— 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 these component XSDs, 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 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.

附錄E

(參考)

XSD實作選擇

主要之實作選擇包含於XSD範例中(參照附錄B)將在E.1至E.5中討論

Annex E

(informative)

XSD implementation choices

Major implementation choices contained in the example XSDs (see Annex B) are discussed in E.1 – E.5.

E.1 資料型式

此組件XSD common/dataTypes.xsd針對資料型式所定義全域的W3C XML架構定義語言型式宣告之收錄，其用以限制定義於第5節中之資料元件的值。提供型式之宣告，以致於在LOM 1.0基礎架構中的邏輯資料型式就其奠基XML架構資料型式而論僅被定義1次。此架構定義之模型化提供可改變奠基架構資料型式之方法，若需要，不須對架構定義做任何其他之改變。

下述全域W3C XML架構定義語言型式定義爲：

(1) CharacterString：XML架構原始資料型式字串之別名。

(2) LanguageId：XML架構衍生資料型式語言之別名。

(3) VCard：CharacterString之別名。

(4) MimeType：CharacterString之別名。

(5) Size：XML架構衍生資料型式nonNegativeInteger之別名。

(6) LanguageString：具選項的定義語言屬性之 0 或多個字串元件所組成之序列。語言屬性描述字串元件擁有值之語言。

(7) DateTime：具選項的子元件dateTime及描述。

(8) Duration：具選項的子元件duration及描述。

LanguageId用以限制多種語言元件之值。

此LanguageString型式允許每一個字串元件包含型式LanguageId之選項屬性語言，其被定義爲XML架構衍生之資料型式語言。

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. 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.
The LanguageString type allows each string element to contain the optional attribute language of type LanguageId, which is defined as the XML Schema derived data type language.

此DateTime型式擁有型式CharacterString之選項的子元件dateTime和型式LanguageString之描述，子元件dateTime之CharacterString型式包含受限制的字元型樣。此型樣爲使用固定表示的W3C XML架構定義語言如下：

([0-9]{3}[1-9] | [0-9]{2}[1-9][0-9] | [0-9][1-9][0-9]{2} | [1-9] [0-9]{3})(\-(0[1-9] | 1[0-2])(\-(0[1-9] | [1-2][0-9] | 3[0-1])(T ([0-1][0-9] | 2[0-3])(:[0-5][0-9](:[0-5][0-9](\.[0-9]{1,}(Z | ((\+|\-)([0-1][0-9] | 2[0-3]):[0-5][0-9]))?)?)?)?)?)?)?

受限制的字元串型樣被用以替代XML架構原始資料型式，因爲值域的限制定義於IEEE 1484.12.1-2002標準中。此限制不能被XML架構原始資料型式dateTime實施。

資料型式Duration具有選項的子元件duration，屬於型式CharacterString，和描述，屬於型式LanguageString。duration子元件之CharacterString型式包含受限制的字元型樣。此型樣爲使用正規表示的W3C XML架構定義語言如下：

P([0-9]{1,}Y){0,1}([0-9]{1,}M){0,1}([0-9]{1,}D){0,1}(T([0-9]{1,}H){0,1}([0-9]{1,}M){0,1}([0-9]{1,}(\.[0-9]{1,}){0,1}S){0,1}) {0,1}

受限制的字元串型樣是用以替代XML架構原始資料型式，因爲值域的限制定義於IEEE 1484.12.1-2002標準中。此限制不能被XML架構原始資料型式duration強制遵守。

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

([0-9]{3}[1-9] | [0-9]{2}[1-9][0-9] | [0-9][1-9][0-9]{2} | [1-9] [0-9]{3})(\-(0[1-9] | 1[0-2])(\-(0[1-9] | [1-2][0-9] | 3[0-1])(T ([0-1][0-9] | 2[0-3])(:[0-5][0-9](:[0-5][0-9](\.[0-9]{1,}(Z | ((\+|\-)([0-1][0-9] | 2[0-3]):[0-5][0-9]))?)?)?)?)?)?)?

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 Std 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, which is defined using a regular expression in W3C XML Schema definition language, is

P([0-9]{1,}Y){0,1}([0-9]{1,}M){0,1}([0-9]{1,}D){0,1}(T([0-9]{1,}

H){0,1}([0-9]{1,}M){0,1}([0-9]{1,}(\.[0-9]{1,}){0,1}S){0,1}) {0,1}

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 Std 1484.12.1-2002. These restrictions cannot be enforced by the XML Schema primitive data type duration.

E.2 元件

組件XSDcommon/elementNames.xsd爲LOM資料元件，定義一個全域W3C XML架構定義語言的資料型式宣告之收錄。對每一個LOM資料元件，此組件XSD包含78個型式宣告。爲其完整性，對元件複製型式宣告，進而分攤包含於XML評論之元件名稱。那些包含於XML評論中之元件應該被留在XML評論內。

提供全域架構型式宣告，以至於合成架構是就其奠基型式之收錄來定義，而非直接以XML架構構成來定義。此架構定義之模型化提供一種方法以改變元件的定義，卻不用對架構定義做任何其他的改變。

對每一個LOM資料元件，XML元件名稱以慣例命名，其採用CamelCase大寫的LOM資料元件命名原則，開頭爲一小寫字母(lowerCamelCase)。(參照第5.4節及第5.5節)

每一個LOM資料元件皆給予全域W3C XML架構定義語言元件之宣告、型式宣告和屬性群組宣告。定義於第5節中之XML名稱用以做每一個宣告之名稱。

典型而論，對元件之W3C XML架構定義語言型式宣告將對此元件提供內容模型，其參考對子元件之全域元件宣告是必要的，且包含從W3C XML架構定義語言屬性群組而來之屬性。LOM 1.0基礎架構聚合元件將有複雜型式呈現之宣告，LOM 1.0基礎架構簡易元件將有簡易型式呈現之宣告。

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 within 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 lowercase letter (*lowerCamelCase*) (see 5.4 and 5.5).

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.

Typically, the W3C XML Schema definition language type declaration for an element will provide the content model for the element, which refers to the global element declarations for subelements as needed and includes 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 聚合

W3C XML架構定義語言結構對聚合元件之選擇方式包括要此聚合元件滿足三個完整的需求，此需求是從LOMv1.0基礎架構而產生直接或間接之需求，需求如下述：

(1) 需求1：IEEE 1484.12.1-2002標準中並沒有定義次序性之限制，每一個元件允許子元件任意排列次序。

(2) 需求2：每一個元件應對其子元件實施多種定義於IEEE 1484.12.1-2002標準之限制。

(3) 需求3：聚合結構應在無介紹額外容器元件下被保留(即元件不直接對應IEEE 1484.12.1-2002定義之元件，但必須要符合前兩項需求)。

The selection of the W3C XML Schema definition language structure for aggregate elements involves satisfying three competing requirements that originate either directly or indirectly from the LOMv1.0 base schema. These requirements are as follows:

— *Requirement 1:* An ordering restriction is not defined in IEEE Std 1484.12.1-2002. Each element should allow arbitrary ordering of its subelements.

— Requirement 2: Each element should enforce multiplicity constraints defined in IEEE Std 1484.12.1-2002 on its subelements.

— Requirement 3: 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 Std 1484.12.1-2002, but they are introduced to meet the first two requirements).

需求1和需求2介紹對聚合元件表達內容模型之複雜性，需求3源自內在限制，其內容模型應盡量對應LOMv1.0基礎結構。W3C XML架構定義語言並不提供可以滿足需求1和需求2且無需求3中介紹額外容器元件之方法。

四個對聚合元件之內容模型的替代性選擇可被實作。此選擇表示於E3.1至E3.4中，爲XML架構處理器之確認方法提供許多選擇。

備考：在E.3.1至E.3.5中，“需求1”、“需求2”和“需求3” 參考子節之需求定義列表。

Requirement 1 and Requirement 2 introduce complexity in expressing the content model for aggregate elements. Requirement 3 originates 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 Requirement 1 and Requirement 2 without the introduction of additional

container elements as in Requirement 3.

Four alternatives for the selection of the content model for aggregate elements could be implemented. These alternatives are presented in E.3.1 – E.3.4. They provide options on the validation approaches taken by XML Schema processors.

NOTE—In E.3.1 – E.3.5, “Requirement 1,” “Requirement 2,” and “Requirement 3” refer to items in the dashed list in this subclause.

E.3.1 使用XML架構元件sequence

XML架構元件sequence之使用提供符合多樣性(需求2)之需求的方法，並避免額外容器元件(需求3)，但不提供任意排序(需求1)。若子元件必須以特定的次序出現，則XSD能簡單的強制遵守無介紹額外內容元件之多樣性限制，圖E.1表示sequence元件如何被用以定義LOM 1.0基礎架構。

圖E.1-使用sequence建構之範例

```
<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>
```

根據圖E.1，general元件可以出現0次或1次(滿足需求2)。此範例並無介紹額外容器元件(滿足需求3)。然而，general元件若出現則必須出現在lifeCycle元件之前(不滿足需求1)。

利用此方法建立之LOM XML實例可能爲不符合的。

The use of the XML Schema element sequence provides a way to meet the requirements of multiplicity (Requirement 2) and avoiding additional container elements (Requirement 3) but does not provide arbitrary ordering (Requirement 1). 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 Requirement 2). The example does not introduce additional container elements (satisfies Requirement 3). However, the general element, if present, has to precede the lifeCycle element (does not satisfy Requirement 1). LOM XML instances created using this approach may be *nonconforming*.

E.3.2 使用XML架構元件all

XML架構元件all提供能夠符合任意排序需求(需求1)之方法。使用all元件需要介紹額外容器元件(需求3)以符合多樣性之需求(需求2)。all元件需要介紹額外容器元件以收錄指定元件之多種實例。LOMv1.0基礎架構包含可具有超過一種多樣性之元件(即教育(Educational))。圖E.2說明all元件如何被用於定義LOM 1.0基礎架構。

圖E.2 使用all元件之範例

```
<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>
```

根據圖E.2，一般(General)元件可以出現0次或1次(滿足需求2)，且可以出現在一序列元件中之任何地方(滿足需求1)。然而，此範例介紹如educationals元件(不滿足需求3)之額外容器元件，複數educationals表示0或多個educationals元件。

利用此方法建立之LOM XML實例可能爲不符合的。

The XML Schema element all provides a way to meet the requirements of arbitrary ordering (Requirement 1). Using the all element requires the introduction of additional container elements (Requirement 3) to meet the requirement of multiplicity (Requirement 2). 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

According to Figure E.2, the general element can appear zero or one time (satisfies Requirement 2) anywhere in the sequence of elements (satisfies Requirement 1). However, the example introduces additional container elements, such as the educationals element (does not satisfy Requirement 3), where the plural educationals indicates zero or more educational elements. LOM XML instances created using this approach may be *nonconforming*.

E.3.3 使用XML架構元件choice

XML架構元件choice之使用提供一方法符合任意排序需求(需求1)和避免額外容器(需求3)，但不強制遵守多樣性限制(需求2)。由允許任一子元件出現多少次，W3C XML架構定義語言之型式可以簡單的建構以允許子元件之任意排序而不必介紹額外容器元件，圖E.3表示choice元件如何用以定義LOM 1.0基礎架構。

圖E.3 使用choice元件之範例

```
<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>
```

在圖E.3之範例中不介紹額外容器元件(滿足需求3)和允許任意排序(滿足需求1)。然而，all元件可能出現任何次數(不滿足需求2)。

此解決方法接受LOMv1.0基礎架構中之超集合，其允許不用介紹LOM XML實例或LOM XSD中之額外複雜度。對願意在PSVI上強制遵守LOM資料元件多樣性限制之應用程式來說，這將呈現一個滿意的解決方法。

利用此方法建立之LOM XML實例可能爲不符合的。

The use of the XML Schema element choice provides a way to meet the requirements of arbitrary ordering (Requirement 1) and avoiding additional containers (Requirement 3) but does not enforce multiplicity constraints (Requirement 2). 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 Requirement 3) and allows arbitrary ordering (satisfies Requirement 1). However, all elements may appear any number of times (does not satisfy Requirement 2).

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

LOM XML instances created using this approach may be *nonconforming*.

E.3.4 使用XML架構元件choice與XML架構限制unique

XML架構元件choice與XML架構限制unique之使用提供符合任意排序(需求1)和避免額外容器(需求3)，以及強制遵守多樣性限制(需求2)之方法。使用此方法，多樣性限制可被使用XML架構限制unique強制遵守。對每一個具有至少一種多樣性在其聚合元件中之元件，其定義一個帶

有相同於元件名稱之固定預設值的uniqueElementName屬性，且使用此屬性爲對unique限制之元件關鍵。圖E.4表示unique限制在當LOMv 1.0基礎架構定義元件不能出現超過1次的時候，可以對元件uniqueness強制遵守多樣性限制。

圖E.4 使用unique限制之範例

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

使用unique此限制，對包含最多一個多樣性之子元件的每一個聚合元件而言是必須的。

此方法的缺點爲要在XSD中介紹uniqueElementName屬性和介紹額外的複雜性。

此解決方式增加了由XML架構處理器而做的效度的總數。對不能處理PSVI之應用程式來說，增加的uniqueElementName屬性可能爲有效的解決方法。

此解決方法支援LOM XML實例符合和完全符合之確認。

The use of the XML Schema element choice with the XML Schema constraint unique provides a way to meet the requirements of arbitrary ordering (Requirement 1) and avoiding additional containers (Requirement 3) and enforces multiplicity constraints (Requirement 2). 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 an XML schema processor. 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 總結

E.3.1至E.3.4中對聚合元件定義選擇性方法，總結如下：

(1) sequence：不滿足需求1，說明子元件可以任意次序性出現。此選擇可能不符合IEEE 1484.12.1-2002標準。

(2) all：不滿足需求3，保留IEEE 1484.12.1-2002定義之聚合架構。使用XML架構元件all需要介紹任意容器元件(即educational)。此選擇需要容器之所有子元件被收錄於聚合元件中。

(3) choicel：不強制遵守唯一性限制，所有元件可能出現任意次數。此選擇不滿足需求2且可能不符合IEEE 1484.12.1-2002標準。此選擇定義於unique/loose.xsd組件XSD中(參照C.2)。

(4) choice with unique：克服與XML架構元件choice之問題。一個屬性可適用於每一個最多有一個多樣性之元件。此屬性藉由應用識別(identity)限制以強制遵守。此選擇定義於unique/strict.xsd組件XSD中(參照C.2)。

Subclauses E.3.1 – E.3.4 defined alternative approaches to aggregate elements. A summary of those alternatives is given as follows:

— sequence: Does not satisfy Requirement 1, which states that the subelements appear in an arbitrary order. This alternative may not conform to IEEE Std 1484.12.1-2002.

— all: Does not satisfy Requirement 3, which preserves the aggregation structure defined in IEEE Std 1484.12.1-2002. 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 elements may appear any number of times. This alternative does not satisfy Requirement 2 and may not conform to IEEE Std 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 詞彙(Vocabulary)

對詞彙(Vocabulary)型式(參照第5.5.5節)之LOM資料元件內容模型的選擇，包括滿足直接或間接由LOMv1.0基礎架構和詞彙如何被應用程式運用所產生的三項要求，此要求如下所述：

(1) 需求1：詞彙(Vocabulary)型式之資料元件應該允許任意值(來源/值 成對方式表示)。

(2) 需求2：若詞彙值之來源爲LOM 1.0基礎架構(即<source>LOMv1.0</source>)，則用以表示詞

彙值之符記定義於第5節中。

(3) 需求3：若詞彙值之來源非LOM 1.0基礎架構(即非<source>LOMv1.0</source>)，則用以表示詞彙值之符記不可與第5節中定義之符記相衝突。

W3C XML架構定義語言無法簡單的爲one-pass模式中之詞彙值提供以完全型式爲目的之檢查。它可以對一組固定的詞彙提供確認，但無法確認任意的詞彙。在LOM XML實例中，固定之詞彙定義於詞彙來源之基礎上。若詞彙的來源有定義，W3C XML架構定義語言建構可定義以幫助詞彙之確認。

W3C XML架構定義語言中表示LOM詞彙的三種選擇呈現於E4.1至E4.3中。此三種選擇提供XML架構處理器確認方法之選擇。

備考：在E.4.1至E.4.3中，“需求1”、 “需求2” 和 “需求3”參考子節中之需求定義列表。

The selection of the content model for LOM data elements of type Vocabulary (see 5.5.5) involves satisfying three requirements that originate either directly or indirectly from the LOMv1.0 base schema and from how vocabularies are to be used by applications. These requirements are as follows:

— *Requirement 1:* Data elements of type Vocabulary should allow arbitrary values (source/value pairs).

— *Requirement 2:* 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.

— Requirement *3:* 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.

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 it 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, “Requirement 1,” “Requirement 2,” and “Requirement 3” refer to the items in the dashed list in this subclause.

E.4.1 vocab/loose.xsd

若對LOM 1.0基礎架構之詞彙值，沒有確認的需求，則可以使用vocab/loose組件XSD。組件XSD提供符合允許任意來源之值(需求1)的方法，但是不確認詞彙值是否爲由來源定義之一組值中有效的一員(需求2和需求3)。此方法對XSD複雜性來說是最簡易的選擇。每一個從詞彙中取值的元件，定義成一個CharacterString，如圖E.5所示。

圖E.5 vocab/loose.組件XSD之範例

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

來源和值兩種元件皆爲選項的，且可以任何次序性出現。對詞彙資料元件沒有任何限制。
此方式接受LOM 1.0基礎架構允許之超集合，但不介紹LOM XML實例中之額外複雜性。此方法的缺點爲不支援詞彙值之確認。使用vocab/loose.xsd組件XSD需要處理PSVI，用以確認用於完全符合和符合之LOM XML實例中的詞彙值。

If no requirements exist for validation of vocabulary values from the LOMv1.0 base schema, 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 (Requirement 1) but does not validate that the vocabulary value is a valid member of the set of values defined by the source (Requirement 2 and Requirement 3). 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

若詞彙只受限於LOM 1.0基礎架構之定義，則可以使用vocab/strict.xsd組件XSD。此組件XSD對LOM 1.0基礎架構之詞彙的確認定義限制。使用此組件XSD需要其來源爲LOM 1.0(即<source>LOMv1.0</source>)，且該值爲有效LOM定義之詞彙(需求2)，但不允許來自任意來源的值(需求1和需求3)。此方法直接實作詞彙元件所允許的值。如圖E.6所示，每個從詞彙取得值之元件定義source元件在哪裡被給定LOM1.0之固定值，和value元件之值來自適當詞彙值之列舉式清單。

圖E.6 vocab/strict.組件XSD之範例

```
<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"/>
<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>
```

source與value兩個元件爲選項的，且可以任意次序性出現。值會以符合定義於第5節中之詞彙符記來確認。

此方法接受LOM 1.0基礎架構所允許的子集合，但是來源不是LOM 1.0基礎架構之詞彙值除外。此方法的優點爲可以確認定義於本標準中之詞彙值。

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 (Requirement 2) but does not allow values from arbitrary sources (Requirement 1 and Requirement 3). 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"/>
<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 the value elements are optional and may appear in any order. Values are validated against the vocabulary tokens defined in Clause 5.

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

爲使用定義在LOM 1.0基礎架構與延伸詞彙中之詞彙，可以使用vocab/custom.xsd組件XSD。此組件XSD對LOM1.0基礎架構之詞彙和定義於其它架構中之詞彙兩者之確認定義限制。使用組件XSD，經由支援其它來源之詞彙值(需求3)和LOM1.0基礎架構詞彙值(需求2)兩者之確認，以提供符合允許任意來源值爲目標(需求1)之方法。此方法提供額外的彈性於標準與延伸詞彙值之部份確認上。

如圖E.7所示，每個從詞彙取得值之元件定義，source元件在哪裡被給定任意值，和value元件之值來自適當詞彙值之慣例清單。

圖E.7 vocab/custom.xsd 組件XSD之範例

```
<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>
</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>
```

source與value元件兩者皆爲選項的，且可以任意次序性出現。值會以符合延伸詞彙例舉式來確

認。

此方法接受LOM 1.0基礎架構所允許之超集合，並且允許使用標準與延伸兩者之詞彙值。此方法的優點爲其保留確認標準詞彙值之能力，同時顧慮到延伸詞彙值的使用。雖然圖E.7之範例中限制定義於LOM1.0基礎架構中當來源爲LOM1.0 (即<source>LOMv1.0</source>)之詞彙值，然而當來源非LOM1.0版時，範例允許LOM1.0基礎架構定義之詞彙值與延伸之詞彙值並用。

備考：若組織計畫延伸詞彙值且遵守本標準中XSD之類似設計，則此組織應該定義名稱空間用以定義延伸詞彙。vocab/custom.xsd 組件XSD爲此延伸(xmlns:lx="http://ltsc.ieee.org/XSD/LOM/custom")提供一保留空間。此組織可以自行變更此宣告。

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 (Requirement 1) by supporting the validation of both vocabulary values from other sources (Requirement 3) and LOMv1.0 base schema vocabulary values (Requirement 2). 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>
</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 the 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 presented 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 額外備考

此小節討論額外的決定因素，用以對LOM1.0基礎架構強制遵守符合性。以下的資訊用於XSD檔案中(Annex B的參考文件)所使用的實作選項。此實作選項包含於dataTypes.xsd組件XSD中。

(1) 某些定義於IEEE 1484.12.1-2002標準中之元件用以表示爲有效之vCard語法。在本標準發行時，並無實際的W3C XML架構定義語言的vCard繫結規格可供使用。XSD實作可能會想定義一型式以封裝VCard型式之元件。VCard資料型式不確認伴隨之良好定義的Internet Mail Consortium 3.0 vCard(參照IETF RFC 2426:1998 [B2])之值。

(2) MimeType全域W3C XML架構定義語言型式，用以定義封裝元件和定義此元件之型式，此元件爲用於儲存MIME型式之元件。MineType型式不確認MimeType值代表有效之MIME型式。(參照IETF RFC 2425:1998 [B2])。

(3) LanguageId型式之元件被建立以封裝語言(Language)元件(參照一般(General)元件，第5.4.5節)之宣告。此型式允許語言(Language)元件擁有語言(Language)識別符(即定義於ISO 639-1、ISO 639-2和ISO 3166-1中)且有大於一之多樣性。

(4) 全域<描述>元件具有LanguageString內容模型和只有1次之多樣性。

(5) 區域元件對描述(Description)元件(參照一般(General)元件，第5.4.1節)與描述(Description)元件(參照教育(Educational)元件，第5.4.5節)之宣告允許此元件有大於1次之多樣性。

(6) 由於描述(Description)元件在LOM 1.0基礎架構中與不同的多樣性出現，因此建立另外兩個XML元件。全域描述元件具有LanguageString內容模型與只有1次之多樣性。全域descriptionUnbounded元件擁有LanguageString內容模型與無限制的多樣性。

(7) 由於source元件既是類路徑(Taxon Path)元件之子元件(參照第5.4.9.2節)又是詞彙(Vocabulary)資料型式(參照第5.5.5節)，而又根據來源(Source)元件之聚合元件而有不同的資料型式，所以在此發展了兩種不同的來源宣告。一種允許類路徑(Taxon Path)元件之確認，另一種允許詞彙值之確認。

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 the XSD

files referenced in Annex B). These implementation choices are included in the dataTypes.xsd component XSD.

— Certain elements are defined in IEEE Std 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 [B2]).

— 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 2425:1998 [B2]).

— An element type of LanguageId has been created to encapsulate the declaration for the Language element (see the General element, 5.4.5). This type allows the Language element to have a Language identifier (as defined in ISO 639-1, ISO 639-2, and ISO 3166-1) and multiplicity greater than one.

— The global <description> element has content model LanguageString 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 Source 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.

英中文名詞對照表

-A-

aggregation	聚合
alias	別名
attribute	屬性

-B-

binding	繫結

-C-

catalog	編目
character	字元
component	組件
constituent	構件
container	容器
context	情境

-D-

-E-

element	元件
entity	個體
entry	*項目*
enumeration	列舉
Extensible Markup Language(XML)	可延伸標示語言(XML)

-F-

-G-

-H-

-I-

identifier	識別符
instance	實例

-J-

-K-

-L-

-M-

makeup	*構成*
metadata	詮釋資料
meta-metadata	詮釋-詮釋資料
-N-	
namespace	名稱空間
-O-	
-P-	
pattern	型樣
-Q-	
-R-	
-S-	
schema	架構
smallest permitted maximum	最小允許上限
string	字串
subelement	子元件
syntax	語法
semantic density	語意密度
-T-	
template	模版
token	符記
taxon	類
-U-	
-V-	
-W-	
-X-	
XML Schema Definition (XSD)	XML 架構定義(XSD)
-Y-	
-Z-	

學習物件詮釋資料之可延伸標示語言架構定義語言繫結 — 英文草案

CNS	Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata	General No..	xxx-x
		Classified No.. .	xxxx-x

Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata

Contents

1. Scope

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 Std 1484.12.1TM-2002.1 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 Std 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, which allows for interoperability and the exchange of LOM XML instances between various systems.

2. Terms and definitions

Definitions and acronyms are defined in 2.1 and 2.2, respectively.

2.1 Definitions

For the purposes of this Standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms* [B1][7] should be referenced for terms not defined in this clause.

2.1.1 aggregate element: A LOM data element that contains other LOM data elements called subelements. *See also:* subelement.

2.1.2 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.

2.1.3 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.

2.1.4 content model: A framework that identifies the makeup (i.e., data types, multiplicity constraints, ordering) of a specific model.

2.1.5 data type: A property of distinct values that indicates the common features of those values and operations on those values.

2.1.6 extended data element: A data element that is not defined in the LOMv1.0 base schema. See also: LOMv1.0 base schema.

2.1.7 extended vocabulary: A value space that is not defined in the LOMv1.0 base schema for a LOM data element of type Vocabulary. See also: LOMv1.0 base schema.

2.1.8 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 wellformed XML document (W3C, XML Information Set). See also: post-schema-validation infoset.

2.1.9 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.

2.1.10 learning object: Any entity, digital or nondigital, that may be used for learning, education, or training. (IEEE 1484.12.1-2002)

2.1.11 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 Std 1484.12.1-2002. See also: LOMv1.0 base schema.

2.1.12 learning object metadata Extensible Markup Language instance (LOM XML instance): A collection of metadata for a learning object that conforms to IEEE Std 1484.12.1-2002, that is represented in XML, and that adheres to the requirements and constraints of the XML Schema binding.

2.1.13 LOMv1.0 base schema: A structured collection of standard data items, including their data types, multiplicities, and container/component relationships, as defined in IEEE Std 1484.12.1-2002, Clause 6.

2.1.14 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)

2.1.15 Multipurpose Internet Mail Extensions type (MIME type): A standard way of classifying content types on the Internet.

2.1.16 post-schema-validation infoset: A transformed version of the XML infoset of a document produced by a conforming 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.

2.1.17 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.

2.1.18 token: A character string. The value space of a token is the set of strings that do not contain the line feed (#xA) or tab (#x9) characters, that have no leading or trailing spaces (#x20), and that have no internal sequences of two or more spaces. The lexical space is the set of strings that does not contain the line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20), and that have no internal sequences of two or more spaces. The base type is normalizedString. Tokens are case sensitive. (Adapted from W3C, XML Schema Part 2). See also: token set.

2.1.19 token set: A set of tokens in which each token is unique. See also: token.

3.1.20 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.

2.1.21 value space: The set of values for a given data type. (ISO/IEC 11404:1996)

NOTE:In IEEE Std 1484.12.1-2002, a value space is typically enumerated outright or defined by reference to another standard or specification.8

2.2 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

3. 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 or corrigenda) applies.

IEEE Std 1484.12.1TM-2002 , IEEE Standard for Learning Object Metadata.2

IETF RFC 2048:1996 , Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures.3

IETF RFC 2426:1998 , vCard MIME Directory Profile.

ISO 639-1 , Code for the Representation of Names of Languages—Part 1: Alpha-2 code.4

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.5

W3C Recommendation (28 October 2004) , XML Schema Part 1: Structures, Second Edition.6

W3C Recommendation (28 October 2004) , XML Schema Part 2: Datatypes, Second Edition.

W3C Recommendation (4 February 2004) , Namespaces in XML 1.1.

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 before 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.

4.1 Strictly conforming LOM XML instances

A *strictly* conforming LOM XML instance

(1) Shall be a strictly conforming LOM instance as defined in IEEE Std 1484.12.1-2002

(2) Shall conform to the requirements of Clause 5

(3) Shall not include vocabulary values that are not defined in Clause 5

(4) Shall not include XML elements or attributes that are not defined in Clause 5

(5) Shall not include mixed content

4.2 Conforming LOM XML instances

A *conforming* LOM XML instance

(1) Shall be a conforming LOM instance as defined in IEEE Std 1484.12.1-2002

(2) Shall conform to the requirements of Clause 5

(3) May include vocabulary values that are not defined in Clause 5

(4) May include XML elements and attributes that are not defined in Clause 5 by using the extension mechanism described in 5.1.3

(5) May include mixed content

5. LOM XML Schema binding definition

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

5.1 General information

As defined in IEEE Std 1484.12.1-2002, all LOM data elements are optional. An 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 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 SPMs as defined in IEEE Std 1484.12.1-2002. The W3C XML Schema definition language does provide a means for restricting maximum lengths (i.e., maxLength for character strings) and maximum numbers of occurrences (i.e., maxOccurs for 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 Std 1484.12.1-2002.)

NOTE:To encourage interoperability, creators of XSDs that intend to validate LOM XML instances against IEEE Std 1484.12.1-2002 should not use maxOccurs or maxLength restrictions or should set the restricted limits 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 Std 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 PSVI of a LOM XML instance is insufficient to determine the validity of the LOM XML instance with respect to IEEE Std 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 Std 1484.12.1-2002, extensions to the LOMv1.0 base schema shall retain the data types and value 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* LOM XML instances:

(1) *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 5.5).

(2) *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).

(3) 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 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 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).

(1) http://ltsc.*ieee*.org/xsd/LOM/custom

(2) http://ltsc.ieee.org/xsd/LOM/unique

(3) http://ltsc.ieee.org/xsd/LOM/vocab

(4) 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.

5.3 Table format and organization

This Standard uses tables to describe the requirements for a LOM XML instance. The tables express requirements for each LOM data element in the LOMv1.0 base schema. These requirements are as follows:

(1) *LOM data* element*:* The name of the LOM data element in the LOMv1.0 base schema.

(2) *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 5.5, 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.

(3) *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, which ensures that the aggregation relationships of the LOMv1.0 base schema components are enforced. The order of appearance of subelements shall not be significant.

(4) *Min:* If a LOM data element is not the top-level LOM element, the requirement 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 requirement on the minimum number of times the LOM data element may appear in a LOM XML instance.

(5) Max: If a LOM data element is not the top-level LOM element, the requirement 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 requirement on the maximum number of times the LOM element may appear in a LOM XML instance. An infinity symbol ("∞") indicates that the maximum number of times is unbounded. Values in parenthesis are SPMs.

(6) 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 "Unspecified." Note: The significance of the order of a list of ordered values is determined by the implementation.

(7) LOM data type: Indicates whether the LOM data type is defined as 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.

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 Std 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 Std 1484.12.1-2002, the values from IEEE Std 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

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
LOM	lom	General Life Cycle Meta-Metadata Technical Educational Rights Relation Annotation Classification	1	1	Unspecified	Unspecified
General	general	See 5.4.1	0	1	Unspecified	Unspecified
Life Cycle	lifeCycle	See 5.4.2	0	1	Unspecified	Unspecified
Meta-Metadata	metaMetadata	See 5.4.3	0	1	Unspecified	Unspecified
Technical	technical	See 5.4.4	0	1	Unspecified	Unspecified
Educational	educational	See 5.4.5	0	∞ (100)	Unordered	Unspecified
Rights	rights	See 5.4.6	0	1	Unspecified	Unspecified
Relation	relation	See 5.4.7	0	∞ (100)	Unordered	Unspecified
Annotation	annotation	See 5.4.8	0	∞ (30)	Unordered	Unspecified
Classification	classification	See 5.4.9	0	∞ (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 is in conformance with “W3C Recommendation, Namespaces in XML 1.1.” Examples are shown in Figure 1 and Figure 2, respectively.

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

. . .

</lom>

Figure 1—An example default 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.

Table 2—The General element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
General	general	Identifier Title Language Description Keyword Coverage Structure Aggregation Level	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.1.1	0	∞ (10)	Unordered	Unspecified
Title	title	See 5.5.4	0	1	Unspecified	LangString
Language	language	None	0	∞ (10)	Unordered	CharacterString
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString
Keyword	keyword	See 5.5.4	0	∞ (10)	Unordered	LangString
Coverage	coverage	See 5.5.4	0	∞ (10)	Unordered	LangString
Structure	structure	See 5.5.5	0	1	Unspecified	Vocabulary
Aggregation Level	aggregationLevel	See 5.5.5	0	1	Unspecified	Vocabulary

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

. . .

</lom:lom>

Figure 2—An example prefix-specific namespace declaration

5.4.1.1 Identifier

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

Table 3—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterString
Entry	entry	None	0	1	Unspecified	CharacterString

5.4.1.2 Language

The Language element shall be a character string. The value of the character string shall be one of the following:

(1) The format and values described in 5.5.4.1

(2) The token none

5.4.1.3 Structure

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

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

(1) atomic

(2) collection

(3) networked

(4) hierarchical

(5) linear

5.4.1.4 Aggregation

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) 1

(2) 2

(3) 3

(4) 4

5.4.2 Life Cycle

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

Table 4—The Life Cycle element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Life Cycle	lifeCycle	Version Status Contribute	0	1	Unspecified	Unspecified
Version	version	See 5.5.4	0	1	Unspecified	LangString
Status	status	See 5.5.5	0	1	Unspecified	Vocabulary
Contribute	contribute	See 5.4.2.2		∞ (30)	Ordered	Unspecified

5.4.2.1 Status

If the value space for the 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:

(1) draft

(2) final

(3) revised

(4) unavailable

5.4.2.2 Contribute

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

Table 5—The Contribute element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type

Contribute	contribute	Role Entity Date	0	∞ (30)	Ordered	Unspecified
Role	role	See 5.5.5	0	1	Unspecified	Vocabulary
Entity	entity	None	0	∞ (40)	Ordered	CharacterString
Date	date	See 5.5.5	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:

(1) author
(2) publisher
(3) unknown
(4) initiator
(5) terminator
(6) validator
(7) editor
(8) graphical designer
(9) technical implementer
(10) content provider
(11) technical validator
(12) educational validator
(13) script writer
(14) instructional designer
(15) 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 Contribute Metadata Schema Language	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.3.1	0	∞ (10)	Unordered	Unspecified
Contribute	contribute	See 5.4.3.2	0	∞ (10)	Ordered	Unspecified
Metadata Schema	metadataSchema	None	0	∞ (10)	Unordered	CharacterStrin g
Language	language	None	0	1	Unspecified	CharacterStrin g

5.4.3.1 Identifier

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

Table 7—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterStrin g
Entry	entry	None	0	1	Unspecified	CharacterStrin g

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 Entity Date	0	∞ (10)	Ordered	Unspecified
Role	role	See 5.5.5	0	1	Unspecified	Vocabulary
Entity	entity	None	0	∞ (10)	Ordered	CharacterStrin g
Date	date	See 5.5.5	0	1	Unspecified	DateTime

5.4.3.2.1 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:

(1) creator

(2) validator

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, 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 instances that are conforming but not strictly conforming (i.e., contain extended data model elements), 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. 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

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Technical	technical	Format Size	0	1	Unspecified	Unspecified
		Location				
		Requirement Installation				
		Remarks				
		Other Platform				
		Requirements Duration				
Format	format	None	0	∞ (40)	Unordered	CharacterString
Size	size	None	0	1	Unspecified	CharacterString
Location	location	None	0	∞ (10)	Ordered	CharacterString
Requirement	requirement	See 5.4.4.3	0	∞ (40)	Unordered	Unspecified
Installation Remarks	installationRemarks	See 5.5.4	0	1	Unspecified	LangString
Other Platform	otherPlatformRequirements	See 5.5.4	0	1	Unspecified	LangString

Requirements						
Duration	duration	See 5.5.3	0	1	Unspecified	Duration

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:

(1) A literal that is the canonical lexical representation of a Multipurpose Internet Mail Extension (MIME) type value from RFC 2048

(2) 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

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Requirement	requirement	OrComposite	0	∞ (40)	Unordered	Unspecified
OrComposite	orComposite	See 5.4.4.3.1	0	∞ (40)	Unordered	Unspecified

5.4.4.3.1 OrComposite

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

Table 11—The OrComposite element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
OrComposite	orComposite	Type Name Minimum Version Maximum Version	0	∞ (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:

(1) operating system

(2) 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:

(1) pc-dos

(2) ms-windows

(3) macos

(4) unix

(5) multi-os

(6) 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 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:

(1) any

(2) netscape communicator

(3) ms-internet explorer

(4) opera

(5) amaya

5.4.5 Educational

Table 12 describes the Educational element and its direct-descendant subelements.

Table 12—The Educational element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type

Educational	educational	Interactivity Type Learning Resource Type Interactivity Level Semantic Density Intended End User Role Context Typical Age Range Difficulty Typical Learning Time Description Language	0	∞ (100)	Unordered	Unspecified
Interactivity Type	interactivityType	See 5.5.5	0	1	Unspecified	Vocabulary
Learning Resource Type	learningResourceType	See 5.5.5	0	∞ (10)	Ordered	Vocabulary
Interactivity Level	interactivityLevel	See 5.5.5	0	1	Unspecified	Vocabulary
Semantic Density	semanticDensity	See 5.5.5	0	1	Unspecified	Vocabulary
Intended End User Role	intendedEndUserRole	See 5.5.5	0	∞ (10)	Ordered	Vocabulary
Context	context	See 5.5.5	0	∞ (10)	Unordered	Vocabulary
Typical Age Range	typicalAgeRange	See 5.5.4	0	∞ (5)	Unordered	LangString
Difficulty	difficulty	See 5.5.5	0	1	Unspecified	Vocabulary
Typical Learning Time	typicalLearningTime	See 5.5.3	0	1	Unspecified	Duration
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString
Language	language	None	0	∞ (10)	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:

(1) active

(2) expositive

(3) 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:

(1) exercise
(2) simulation
(3) questionnaire
(4) diagram
(5) figure
(6) graph
(7) index
(8) slide
(9) table
(10) narrative text
(11) exam
(12) experiment
(13) problem statement
(14) self assessment
(15) 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:

(1) very low
(2) low
(3) medium
(4) high
(5) 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:

(1) very low
(2) low
(3) medium
(4) high

(5) 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:

(1) teacher

(2) author

(3) learner

(4) manager

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:

(1) school

(2) higher education

(3) training

(4) 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:

(1) very easy

(2) easy

(3) medium

(4) difficult

(5) very difficult

5.4.5.8 Language

The Language element shall be a character string. 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.

Table 13—The Rights element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Rights	rights	Cost Copyright And Other Requirements Description	0	1	Unspecified	Unspecified
Cost	cost	See 5.5.5	0	1	Unspecified	Vocabulary

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:

(1) yes

(2) 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 of tokens:

(1) yes

(2) no

5.4.7 Relation

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

Table 14—The Relation element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Relation	relation	Kind Resource	0	∞ (100)	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:

(1) ispartof

(2) haspart

(3) isversionof

(4) hasversion

(5) isformatof

(6) hasformat

(7) references

(8) isreferencedby

(9) isbasedon

(10) isbasisfor

(11) requires

(12) isrequiredby

5.4.7.2 Resource

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

Table 15—The Resource element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Resource	resource	Identifier Description	0	1	Unspecified	Unspecified
Identifier	identifier	See 5.4.7.2.1	0	∞ (10)	Unordered	Unspecified
Description	description	See 5.5.4	0	∞ (10)	Unordered	LangString

5.4.7.2.1 Identifier

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

Table 16—The Identifier element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Identifier	identifier	Catalog Entry	0	∞ (10)	Unordered	Unspecified
Catalog	catalog	None	0	1	Unspecified	CharacterStrin g
Entry	entry	None	0	1	Unspecified	CharacterStrin g

5.4.8 Annotation

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

Table 17—The Annotation element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Annotation	annotation	Entry Date Description	0	∞ (30)	Unordered	Unspecified
Entity	entity	None	0	1	Unspecified	CharacterStrin g
Date	date	See 5.5.2	0	1	Unspecified	DateTime
Description	description	See 5.5.4	0	1	Unspecified	LangString

5.4.8.1 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.

5.4.9 Classification

Table 18 describes the Classification element and its direct-descendant subelements.

Table 18—The Classification element

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Classification	classification	Purpose Taxon Path Description Keyword	0	∞ (40)	Unordered	Unspecified
Purpose	purpose	None	0	1	Unspecified	Vocabulary
Taxon Path	taxonPath	See 5.4.9.2	0	∞ (15)	Unordered	Unspecified
Description	description	See 5.5.4	0	1	Unspecified	LangString
Keyword	keyword	See 5.5.4	0	∞ (40)	Ordered	LangString

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:

(1) discipline

(2) idea

(3) prerequisite

(4) educational objective

(5) accessibility restrictions

(6) educational level

(7) skill level

(8) security level

(9) 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 Taxon	0	∞ (15)	Unordered	Unspecified
Source	source	See 5.5.4	0	1	Unspecified	LangString
Taxon	taxon	See 5.4.9.2.1	0	∞ (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

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Taxon	taxon	Id Entity	0	∞ (15)	Ordered	Unspecified
Id	id	None	0	1	Unspecified	CharacterString
Entry	entry	See 5.5.4	0	1	Unspecified	LangString

5.5 Common data types and elements

Data types that are used in the LOM XML 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:

(1) DateTime

(2) Description

If present, these subelements 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 element	XML name	Subelements	Min	Max	Order	LOM data type
DateTime	—	DateTime Description	0	1	Unspecified	Unspecified
DateTime	dateTime	None	0	1	Unspecified	CharacterString
Description	description	See 5.5.4	0	1	Unspecified	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: YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]]]

Where YYYY is the four-digit year (≥0001)

MM is the two-digit month (01 through 12, where 01 = January, etc.)

DD is the two-digit day of a month (01 through 31, depending on the values of month and year)

hh are the two digits of an hour (00 through 23) (AM and PM are NOT allowed)
mm are the wo digits of a minute (00 through 59)
ss are the two digits of a second (00 through 59)
s is one or more digits representing a decimal fraction of a second
TZD is the time zone designator (“Z” for coordinated universal time [UTC] or +hh, -hh, +hh:mm, or -hh:mm, where hh is two digits of an hour and mm is two digits of a 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.

NOTE 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 Std 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 before 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 “]”) indicate optional elements that may appear zero or one time in the character lexical representation of the DateTime element. These meta characters do not appear in the result; only the associated values described appear (e.g., DD is replaced by the corresponding two-digit value for day of month).

4.The value space is based on ISO 8601:2000 [B3].

5.5.3 Duration

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:

(1) Duration
(2) Description

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

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

Table 22—The Duration data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Duration	—	Duration Description	0	1	Unspecified	Unspecified
Duration	duration	None	0	1	Unspecified	CharacterStrin g

Description	description	See 5.5.4	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[yY][mM][dD][T[hH][nM][s[.s]S]] where

y is the number of years (integer, ≥ 0)

m is the number of months (integer, ≥ 0, not restricted, e.g., >12 is acceptable)

d is the number of days (integer, ≥ 0, not restricted, e.g., >31 is acceptable)

h is the number of hours (integer, ≥ 0, not restricted, e.g., >23 is acceptable)

n is the number of minutes (integer, ≥ 0, not restricted, e.g., >59 is acceptable)

s is the number of seconds or fraction of seconds (integer, ≥0, not restricted, e.g., >59 is acceptable)

The character literal designators “P”, “Y”, “M”, “D”, “T”, “H”, “M”, and “S” 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 integer value shall be present in addition to the designator “P”. The designator “T” shall be omitted if all of the time (hours, minutes, and seconds) is zero. Negative durations are not supported.

NOTE 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 Std 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 designator “D”).

5.The value space is based on ISO 8601:2000 [B3].

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:

(1) String

(2) Language

If present, these subelements 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

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
LangString	—	String Language	0	(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:

(1) Two-letter codes are defined in ISO 639-1.

(2) Three-letter codes are defined in ISO 639-2.

(3) The one-letter code “i” is reserved and uses as a prefix for registrations defined by the IANA.

(4) The one-letter code “x” is reserved and used as a prefix for private use.

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

(6) Two-letter subcodes are ISO 3166-1 alpha-2 country codes.

(7) Subcodes from three to eight characters in length 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.

NOTE 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 constraints on this language code.

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:

(1) Source

(2) Value

If present, these subelements 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 24 describes the Vocabulary data type and its direct-descendant subelements.

Table 24—The Vocabulary data type

LOM data element	XML name	Subelements	Min	Max	Order	LOM data type
Vocabulary	—	Source 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.

NOTE:The XML Schema derived data type token supports the repertoire of ISO/IEC 10646-1. The token data type ensures the support of multiple languages including multibyte languages. 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.

Annex A

(informative)

Bibliography

[B1] IEEE 100TM, The Authoritative Dictionary of IEEE Standards Terms.9

[B2] IETF RFC 2425:1998, MIME Content-Type for Directory Information.10

[B3] ISO 8601:2000, Data Elements and Interchange Formats—Information Interchange—Representation of Dates and Times.11

[B4] ISO/IEC 11404:1996, Information Technology—Programming Languages, Their Environments and System Software Interfaces—Language-Independent Datatypes.12

[B5] W3C Recommendation (4 February 2004), Extensible Markup Language (XML) 1.1.13

[B6] W3C Recommendation (4 February 2004), XML Information Set, Second Edition.

Annex B

(informative)

Internet availability of XSD files

XSD files are available as examples on the World Wide Web; however, they are not to be construed as a part of this Standard, nor endorsed by the IEEE for any specific purpose. USE AT YOUR OWN RISK.

The XSD files are available at the following URL:

http://standards.ieee.org/reading/ieee/downloads/LOM/lomv1.0/

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

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 it 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.

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 the PSVI. 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 *conforming* or *strictly conforming* LOM instance has to conform to the rules defined in Clause 5.

Informative W3C XML XSDs are available for implementers to satisfy many requirements of IEEE Std 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 and in 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 and in 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 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 LOM XML Schema binding constraints 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 Std 1484.12.1-2002.

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.)

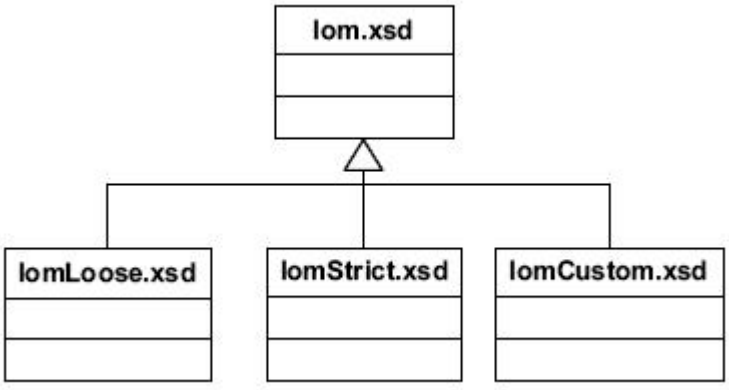


Figure C.1—The relationship of the composite XSDs

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).

C.1.1 lomStrict.xsd

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 Std 1484.12.1-2002. The lomStrict.xsd composite XSD

(1) Limits vocabularies (source/value pairs) to those defined in Clause 5

(2) Expresses the uniqueness constraints defined in Clause 5

(3) Does not permit extensions to the LOMv1.0 base schema (see 5.1.3)

C.1.2 lomCustom.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 Std 1484.12.1-2002. The lomCustom.xsd composite XSD

(1) Defines a mechanism to describe vocabularies (source/value pairs) in addition to those defined in Clause 5. An XML schema processor 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.
(2) Expresses the uniqueness constraints defined in Clause 5.
(3) 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. An XML schema processor 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 lomLoose.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 Std 1484.12.1-2002. The lomLoose.xsd composite XSD

(1) Does not provide validation to any defined vocabulary values (i.e., values defined in Clause 5 or values defined in an extended vocabulary). An XML schema processor will verify that the vocabulary values are valid tokens only.
(2) Does not express the uniqueness constraints defined in Clause 5.
(3) 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. An XML schema processor 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. Applications that use the vocab/loose.xsd component XSD should enforce those vocabulary requirements by other means.

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:

(1) common/anyElement.xsd: Defines the base XML Schema model groups for elements and attributes used for extended data elements and XML attributes.
(2) common/dataTypes.xsd: Defines global data type declarations for data types defined in Clause 5.
(3) common/elementNames.xsd: Defines global element declarations for each data element defined in Clause 5. 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.
(4) common/elementTypes.xsd: Defines global data type declarations for data elements defined in Clause 5. 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.
(5) common/rootElement.xsd: Defines the element name declaration for the element that contains all other LOM data elements for a LOM XML instance.
(6) common/vocabTypes.xsd: Defines global data type declarations for those LOM data elements that

have values taken from a vocabulary data type defined in Clause 5.

(7) common/vocabValues.xsd: Defines the standard vocabulary value declarations defined in Clause 5. These declarations are used in conjunction with both vocab/custom.xsd and vocab/strict.xsd.

(8) 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, by using the extend/custom.xsd component XSD. LOM XML instances that use extensions will *conform* but not *strictly conform* to this Standard.

(9) extend/strict.xsd: 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 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.

(10) unique/loose.xsd: Defines XML Schema model group declarations 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. Applications that use of the unique/loose.xsd component XSD have to enforce those uniqueness constraints by other means.

(11) unique/strict.xsd: Defines XML Schema model group declarations for LOM data elements defined in Clause 5 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.

(12) vocab/custom.xsd: Enforces that vocabulary values are only those specified in Clause 5 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 and extended vocabulary values using the vocab/custom.xsd component XSD is desirable under some circumstances, but it may complicate the schema validation process.

(13) 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 nonconforming. Applications that require validation of the source/value pairs will have to process the PSVI. *Note:* The

absence of the enforcement of 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. Applications that use the vocab/ loose.xsd component XSD have to enforce those vocabulary constraints by other means.

(14) vocab/strict.xsd: Enforces that vocabulary values are only those specified in Clause 5. This component XSD does not allow extended vocabulary 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.

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:

(1) Extensions to the LOMVv1.0 base schema retain the value spaces and data types of LOM data elements from the LOMv1.0 base schema.

(2) 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.

NOTE 1.To ensure that conforming LOM XML instances are created, an organization that develops extensions also should 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 organizationdefines a namespace, then the XSDs referenced in Annex B 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 may 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 referenced in 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 as follows:

(1) common/anyElement.xsd: Defines the base XML Schema model group for elements used for extended data elements.

(2) common/dataTypes.xsd and common/elementTypes.xsd: For each aggregate complexType declaration defined in these component XSDs, an XML Schema model group reference to the customElements declaration is defined.

(3) 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.

(4) 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 may 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 as follows:

(1) common/anyElement.xsd: Defines the base XML Schema model group for elements used for extended data elements.

(2) common/dataTypes.xsd and common/elementTypes.xsd: For each aggregate complexType declaration defined in these component XSDs, an XML Schema model group reference to the customAttributes declaration is defined.

(3) common/vocabTypes.xsd: For each vocabulary and value complexType declaration defined in this component XSD, an XML Schema attributeGroup reference to the customAttributes XML Schema model group declaration is defined.

(4) 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 Annex B) are discussed in E.1 – E.5.

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. 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:

(1) CharacterString: An alias for the XML Schema primitive data type string.

(2) LanguageId: An alias for the XML Schema derived data type language.

(3) VCard: An alias for CharacterString.

(4) MimeType: An alias for CharacterString.

(5) Size: An alias for the XML Schema derived data type nonNegativeInteger.

(6) 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.

(7) DateTime: Optional subelements dateTime and description.

(8) Duration: Optional subelements duration and description.

LanguageId is used to constrain the values of the various language elements.

The LanguageString type allows each string element to contain the optional attribute language of type LanguageId, which is defined as the XML Schema derived data type language.

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

```
([0-9]{3}[1-9] | [0-9]{2}[1-9][0-9] | [0-9][1-9][0-9]{2} | [1-9] [0-
9]{3})(\-(0[1-9] | 1[0-2])(\-(0[1-9] | [1-2][0-9] | 3[0-1])(T ([0-1][0-
9] | 2[0-3])(:[0-5][0-9](:[0-5][0-9](\.[0-9]{1,}(Z | ((\+|\-)([0-1][0-9]
| 2[0-3]):[0-5][0-9]))?)?)?)?)?)?)?
```

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 Std 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, which is defined using a regular expression in W3C XML Schema definition language, is

```
P([0-9]{1,}Y){0,1}([0-9]{1,}M){0,1}([0-9]{1,}D){0,1}(T([0-9]{1,}
```

H){0,1}([0-9]{1,}M){0,1}([0-9]{1,}(\.[0-9]{1,}){0,1}S){0,1}) {0,1}

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 Std 1484.12.1-2002. These restrictions cannot be enforced by the XML Schema primitive data type duration.

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 within 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 lowercase letter (*lowerCamelCase*) (see 5.4 and 5.5).

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.

Typically, the W3C XML Schema definition language type declaration for an element will provide the content model for the element, which refers to the global element declarations for subelements as needed and includes 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 originate either directly or indirectly from the LOMv1.0 base schema. These requirements are as follows:

(1) Requirement *1:* An ordering restriction is not defined in IEEE Std 1484.12.1-2002. Each element should allow arbitrary ordering of its subelements.

(2) Requirement *2:* Each element should enforce multiplicity constraints defined in IEEE Std 1484.12.1-2002 on its subelements.

(3) Requirement *3:* 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 Std 1484.12.1-2002, but they are introduced to meet the first two requirements).

Requirement 1 and Requirement 2 introduce complexity in expressing the content model for aggregate elements. Requirement 3 originates 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 Requirement 1 and Requirement 2 without the introduction of additional container elements

as in Requirement 3.

Four alternatives for the selection of the content model for aggregate elements could be implemented. These alternatives are presented in E.3.1 – E.3.4. They provide options on the validation approaches taken by XML Schema processors.

NOTE:In E.3.1 – E.3.5, “Requirement 1,” “Requirement 2,” and “Requirement 3” refer to items in the dashed list in this subclause.

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 (Requirement 2) and avoiding additional container elements (Requirement 3) but does not provide arbitrary ordering (Requirement 1). 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 Requirement 2). The example does not introduce additional container elements (satisfies Requirement 3). However, the general element, if present, has to precede the lifeCycle element (does not satisfy Requirement 1).

LOM XML instances created using this approach may be *nonconforming*.

E.3.2 Using the XML Schema element all

The XML Schema element all provides a way to meet the requirements of arbitrary ordering (Requirement 1). Using the all element requires the introduction of additional container elements (Requirement 3) to meet the requirement of multiplicity (Requirement 2). 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

According to Figure E.2, the general element can appear zero or one time (satisfies Requirement 2) anywhere in the sequence of elements (satisfies Requirement 1). However, the example introduces additional container elements, such as the educationals element (does not satisfy Requirement 3), where the plural educationals indicates zero or more educational elements. LOM XML instances created using this approach may be *nonconforming*.

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 (Requirement 1) and avoiding additional containers (Requirement 3) but does not enforce multiplicity constraints (Requirement 2). 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 Requirement 3) and allows arbitrary ordering (satisfies Requirement 1). However, all elements may appear any number of times (does not satisfy Requirement 2).

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

LOM XML instances created using this approach may be *nonconforming*.

E.3.4 Using the XML Schema element choice with the XML Schema constraint unique

The use of the XML Schema element choice with the XML Schema constraint unique provides a way to meet the requirements of arbitrary ordering (Requirement 1) and avoiding additional containers (Requirement 3) and enforces multiplicity constraints (Requirement 2). 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 an XML schema processor. 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 as follows:

(1) sequence: Does not satisfy Requirement 1, which states that the subelements appear in an arbitrary order. This alternative may not conform to IEEE Std 1484.12.1-2002.

(2) all: Does not satisfy Requirement 3, which preserves the aggregation structure defined in IEEE Std 1484.12.1-2002. 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.

(3) choice: Does not enforce uniqueness constraints. All elements may appear any number of times. This alternative does not satisfy Requirement 2 and may not conform to IEEE Std 1484.12.1-2002. This alternative is defined in the unique/loose.xsd component XSD (see C.2).

(4) 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 (see 5.5.5) involves satisfying three requirements that originate either directly or indirectly from the LOMv1.0 base schema and from how vocabularies are to be used by applications. These requirements are as follows:

(1) Requirement *1:* Data elements of type Vocabulary should allow arbitrary values (source/value pairs).

(2) Requirement *2:* 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.

(3) Requirement *3:* 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.

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 it 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, “Requirement 1,” “Requirement 2,” and “Requirement 3” refer to the items in the dashed list in this subclause.

E.4.1 vocab/loose.xsd

If no requirements exist for validation of vocabulary values from the LOMv1.0 base schema, 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 (Requirement 1) but does not validate that the vocabulary value is a valid member of the set of values defined by the source (Requirement 2 and Requirement 3). 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 (Requirement 2) but does not allow values from arbitrary sources (Requirement 1 and Requirement 3). 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"/>
<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 the value elements are optional and may appear in any order. Values are validated against the vocabulary tokens defined in Clause 5.

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 (Requirement 1) by supporting the validation of both vocabulary values from other sources (Requirement 3) and LOMv1.0 base schema vocabulary values (Requirement 2). 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>
</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 the 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 presented 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 the XSD files referenced in Annex B). These implementation choices are included in the dataTypes.xsd component XSD.

(1) Certain elements are defined in IEEE Std 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 [B2]).

(2) 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 2425:1998 [B2]).

(3) An element type of LanguageId has been created to encapsulate the declaration for the Language element (see the General element, 5.4.5). This type allows the Language element to have a Language identifier (as defined in ISO 639-1, ISO 639-2, and ISO 3166-1) and multiplicity greater than one.

(4) The global <description> element has content model LanguageString and multiplicity one.

(5) 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.

(6) 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.

(7) Because the Source 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.

學習物件詮釋資料之可延伸標示語言架構
定義語言繫結
— 參考資料、爭議事項、英中名詞對照

1. 參考資料

IEEE 100TM, The Authoritative Dictionary of IEEE Standards Terms.

IETF RFC 2425:1998, MIME Content-Type for Directory Information.

ISO 8601:2000, Data Elements and Interchange Formats—Information Interchange—Representation o f Dates and Times.

ISO/IEC 11404:1996, Information Technology—Programming Languages, Their Environments and System Software Interfaces—Language-Independent Datatypes.

W3C Recommendation (4 February 2004), Extensible Markup Language (XML) 1.1.

W3C Recommendation (4 February 2004), XML Information Set, Second Edition

2. 爭議事項

無

3. 英中名詞對照表

-A-

aggregation	聚合
alias	*別名*
attribute	屬性

-B-

binding	繫結

-C-

catalog	編目
character	字元
component	組件
constituent	構件
container	容器
context	*情境*

-D-

-E-

element	元件
entity	實體
entry	*款目*
enumeration	列舉
Extensible Markup Language(XML)	*可延伸標示語言(XML)*

-F-

-G-

-H-

-I-

identifier	識別符
instance	實例

-J-

-K-

-L-

LangString	*語言字串*

-M-

metadata	後設資料
meta-metadata	*後設-後設資料*

-N-

namespace	名稱空間

-O-

-P-

pattern	型樣

-Q-

-R-

-S-

schema	架構
set	集合
smallest permitted maximum	最小允許上限
string	字串
subelement	子元件
syntax	語法
semantic density	語意密度

-T-

template	模板

token	符記
taxon	*類*
-U-	
-V-	
-W-	
-X-	
XML Schema Definition（XSD）	*XML 結構定義(XSD)*
-Y-	
-Z-	

4. 中英名詞對照表

聚合	aggregation
別名	*alias*
屬性	attribute
繫結	binding
編目	catalog
字元	character
組件	component
構件	constituent
容器	container
情境	*context*
元件	element
實體	entity
款目	*entry*
列舉	enumeration
可延伸標示語言(XML)	*Extensible Markup Language(XML)*
識別符	identifier
實例	instance
語言字串	*LangString*
後設資料	metadata

後設-後設資料	*meta-metadata*
名稱空間	namespace
型樣	pattern
架構	schema
集合	set
最小允許上限	smallest permitted maximum
字串	string
子元件	subelement
語法	syntax
語意密度	semantic density
模板	template
符記	token
類	*taxon*
XML 結構定義(XSD)	*XML Schema Definition (XSD)*