教材物件溝通資料模型之可延伸標示語言架構繫結 — 中文草案

中華民國國家標準 CNS	教材物件溝通資料模型之可延伸標示語言架構繫結	總號..	xxxxx-x
		類號..	xxxx-x

Extensible Markup Language (XML) Schema Binding for Data Model for Content Object Communication

目錄

1.適用範圍

1.1 範圍

本標準係爲IEEE 1484.11.1TM-2004.([1])標準中全球資訊網聯盟(World Wide Web Consortiu, W3C)可延伸標示語言(XML)架構繫結的資料模型。符合此標準的實作應同時符合IEEE 1484.11.1TM-2004.1 標準。

註([1]) 參見第二節。

This Standard specifies a World Wide Web Consortium (W3C) Extensible Markup Language (XML) Schema binding of the data model defined in IEEE Std 1484.11.1TM-2004([1]). An implementation that conforms to this Standard shall conform to IEEE Std 1484.11.1-2004.

1.2 目標

本標準旨在允許以 XML 產生 IEEE 1484.11.1-2004 標準的資料模型實例，使用 W3C XML 架構定義語言以規定此資料模型實例的編碼(參照 XML 架構第 1 部和第 2 部)能在各種系統間互運及交換之資料模型實例。

The purpose of this Standard is to allow the creation of IEEE Std 1484.11.1-2004

data-model instances in XML. This Standard uses the W3C XML Schema definition language to specify the encoding of these data-model instances (see XML Schema Parts 1 and 2), which allows for interoperability and the exchange of data-model instances between various systems.

2.用語釋義

2.1 定義

下述係用語及其定義，其中未定義者可參照IEEE官方辭典的標準辭彙 (The Authoritative Dictionary of IEEE Standards Terms) ([1])。

2.1.1 教材物件(content object)：學習技術系統呈現給學習者大量的數位內容，可能包括學習素材和處理碼。例如：教材物件可能是具有嵌入式影像剪輯及ECMAScript的的互動HTML網頁([1])。

2.1.2 教材物件溝通資料之可延伸標示語言實例(COCD XML 實例)：IEEE 1484.11.1-2004 標準定義，以特定 XML 表示的資料模型，具有 XML 繫結資料模型的條件和限制。

2.1.3 可延伸標示語言繫結(XML 繫結)(Extensible Markup Language binding (XML binding))：係 W3C 可延伸標示語言中資料模型元件之行爲、屬性、值空間的編碼方法。本方法使用 W3C XML 架構定義語言。

註([1]) 對照附錄A的參考文獻。

2.2 縮寫

COCD	教材物件溝通資料(content object communication data)
SPM	最小允許上限值(smallest permitted maximum)
W3C	全球資訊網聯盟(World Wide Web Consortium)
XML	可延伸標示語言(Extensible Markup Language)
XSD	XML 架構定義(XML Schema definition)

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For 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.

3.1.1content object: A collection of digital content that is intended for presentation to a learner by a learning technology system. A content object may include learning material and processing code. Example: A content object might be an interactive HTML page with an embedded video clip and an ECMAScript.

3.1.2 content object communication data Extensible Markup Language instance

(COCD XML instance): A particular XML representation of the data model defined in IEEE Std 1484.11.1-2004 that adheres to the requirements and constraints of an XML binding of the data model.

3.1.3 Extensible Markup Language binding (XML binding): The method of encoding the behaviors, attributes, and value spaces of data-model elements in W3C Extensible Markup Language. This method is specified using the W3C XML Schema definition language.

3.2 Acroynms and abbreviations

COCD	content object communication data
SPM	smallest permitted maximum
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，僅引版本爲證；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) applies.

IEEE 1484.11.1-2004 IEEE學習技術標準－教材物件溝通用資料模型([2]) (Data Model for Content Object Communication)

W3C建議(2004 年 10 月 28 日) XML架構第 1 部分：結構,第二版([3]) (XML Schema Part 1: Structures, Second Edition

W3C 建議(2004 年 10 月 28 日) XML 架構第 2 部分：資料型式,第二版(XML Schema Part 2: Datatypes, Second Edition)

註([2]) 此處提及的IEEE標準或出版品皆以IEEE商標標明。

([3]) W3C(http://www.w3.org/).

IEEE Std 1484.11.1-2004, IEEE Standard for Learning Technology—Data Model for Content Object Communication.2

W3C Recommendation (28 October 2004), XML Schema Part 1: Structures, Second Edition.3

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

4.符合性

本標準以XML繫結定義符合IEEE 1484.11.1-2004標準之教材物件溝通資料實例。此後，此類實例被稱爲「COCD XML實例」。

本標準中，「應」指實作的必要條件；「不應」則指禁止。

This Standard defines conforming IEEE Std 1481.11.1-2004 content object communication data (COCD) instances in an XML binding. Hereafter, such instances are referred to as “COCD XML instances.”

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

符合 COCD XML 的實例

(1)應符合 IEEE 1484.11.1-2004 標準資料模型需求。

(2)不應包含 IEEE 1484.11.1-2004 標準定義之資料模型的任何延伸。

(3)應能以附錄 B 中載明的 XML 架構定義(XSD)驗證。

(4)不應包含任何附錄 B 中 XSD 未定義的元件或屬性。

(5)應由單一元件和它的後代所構成。此單一元件應有附錄 B 中 XSD 定義的名稱〝cocd〞，且應存使用第 5 節中規定之名稱領域所宣告之名稱領域範圍內。

備考 1.本標準不需將 COCD XML 實例完成爲一份 XML 文件，它可能會以深藏其他名稱領域元件的 XML 資料實例。

2.IEEE1484.11.1-2004 標準定義了最小允許上限值(SPM)。如果 COCD XML 實例包含多於 COCD 元件出現的 SPM 次數，則實作者應察覺無法保證，應用能處理多於 COCD 元件出現的 SPM 次數。如果 COCD XML 實例超過多個字元串的 SPM 字元值，則實作者應當有無法處理多個字元串的 SPM 字元值之認知。

3.W3C XML 架構定義語言不能表達和強制所有 IEEE 1484.11.1-2004 標準資料模型之需求(例如，SPM 之需求)。

A conforming COCD XML instance

— Shall conform to the data-model requirements of IEEE Std 1484.11.1-2004.

— Shall not contain any extensions to the data model defined in IEEE Std 1484.11.1-2004.

— Shall be valid according to the XML Schema definition (XSD) specified in Annex B.

— Shall not contain any elements or attributes not defined in the XSD specified in Annex B.

— Shall consist of a single element and its descendants. The single element shall have the name “cocd” as defined in the XSD specified in Annex B. The single element shall reside within the scope of a namespace declaration using the namespace specified in Clause 5.

NOTE 1—This Standard does not require that the COCD XML instance be an XML document. The instance may be embedded at any depth in an XML data instance that includes elements from other name-spaces. 5

NOTE 2—IEEE Std 1484.11.1-2004 defines smallest permitted maximum (SPM) values. If a COCD XML instance contains more than the SPM number of occurrences of a COCD element, implementers should be aware that it is not guaranteed that an application will process more than the SPM number of occurrences of the COCD element. If a COCD XML instance contains more than the SPM number of characters in a character string, implementers should be aware that it is not guaranteed that an application will process more than the SPM number of characters in the character string.

NOTE 3—The W3C XML Schema definition language cannot express and enforce all data-model requirements of IEEE Std 1484.11.1-2004 (e.g., the requirements for SPMs).

5.XML 繫結

XML 繫結的名稱領域係以符合附錄 B 的 XSD 所定義（係爲 http://ltsc.ieee.org/xsd/1484_11_3）。

附錄 B 中的 XSD 符合 XML 架構的第 1 部份和第 2 部分(2004 年 10 月)。

附錄 C 中會提供 COCD XML 實例的範例。

備考：建議XSD的檔案名稱爲〝ieee_1484_11_3_2005.xsd〞，此名稱會被視爲一預設檔名，不應用於任何檔案命名，除非符合是附錄B定義的XSD。

The namespace for the XML binding is defined by the conforming XSD in Annex B and shall be http://ltsc.ieee.org/xsd/1484_11_3

The XSD in Annex B conforms to XML Schema Parts 1 and 2, October, 2004.

An example COCD XML instance is given in Annex C.

NOTE—The recommended file name for the XSD is “ieee_1484_11_3_2005.xsd”. This file name should be treated as a reserved file name; it should not be used to name any file other than the conforming XSD defined in Annex B.

附錄 A

(參考)

參考文獻

[B1] IEEE 100, IEEE官方辭典的標準辭彙，第七版([1])。

註([1]) http://standards.ieee.org/

[B2] ISO/IEC 11404:1996資訊技術－程式語言、運作環境與軟體介面–語言獨立資料型式。

Annex A

(informative)

Bibliography

[B1] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.6

[B2] ISO/IEC 11404:1996, Information technology—Programming languages, their environments and system software interfaces—Language-independent datatypes.7

附錄 B

(規定)

XSD 規定

圖 B.1 顯示符合 IEEE 1484.11.1-2004 標準定義之資料模型 XSD。

Annex B

(normative)

Normative XSD

Figure B.1 shows the conforming XSD for the data model defined by IEEE Std 1484.11.1-2004.

圖 B.1—符合的 XSD

Figure B.1—Conforming XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns="http://ltsc.ieee.org/xsd/1484_11_3"
   xmlns:t="http://ltsc.ieee.org/xsd/1484_11_3"
   targetNamespace="http://ltsc.ieee.org/xsd/1484_11_3"
   elementFormDefault="qualified" version="1484.11.3-1.0">
   <xs:annotation>
      <xs:documentation xml:lang="en">
        This schema is specified in IEEE 1484.11.3-2005, "IEEE Standard
        for Learning Technology - Extensible Markup Language (XML)
        Binding for Data Model for Content Object Communication."
        This schema is a World Wide Web Consortium (W3C) Extensible
        Markup Language (XML) binding of the data model defined in IEEE
        1484.11.1-2004, "IEEE Standard for Learning Technology - Data
        Model for Content Object Communication."
        The purpose of this schema is to allow the creation of IEEE
        1418.11.1-2004 data-model instances in XML. This schema uses the
        W3C XML Schema definition language as the encoding. This allows
        for interoperability and the exchange of data-model instances
        between various systems.
        This schema shall not be modified but may be included in
        derivative works.

        Copyright (c) 2005 Institute of Electrical and Electronics
        Engineers, Inc.
```

```
            USE AT YOUR OWN RISK


    </xs:documentation>
</xs:annotation>
<!-- -->
<!-- The first half of this document follows the order of the
      Data Model document clauses. -->
<!-- -->
<xs:element name="cocd" type="cocdType"/>
<xs:complexType name="cocdType">
   <xs:annotation>
          <xs:documentation xml:lang="en">
              Implements IEEE 1484.11.1-2004,
              Clause 6.1: Content object communication
          </xs:documentation>
     </xs:annotation>
     <xs:all>
       <xs:element ref="commentsFromLearner" minOccurs="0"/>
        <xs:element ref="commentsFromLMS" minOccurs="0"/>
        <xs:element ref="completionStatus" minOccurs="0"/>
        <xs:element ref="completionThreshold" minOccurs="0"/>
        <xs:element ref="credit" minOccurs="0"/>
        <xs:element ref="dataModelVersion" minOccurs="0"/>
        <xs:element ref="entry" minOccurs="0"/>
        <xs:element ref="exit" minOccurs="0"/>
        <xs:element ref="interactions" minOccurs="0"/>
        <xs:element ref="launchData" minOccurs="0"/>
        <xs:element ref="learnerId" minOccurs="0"/>
        <xs:element ref="learnerName" minOccurs="0"/>
        <xs:element ref="learnerPreferenceData" minOccurs="0"/>
        <xs:element ref="lessonStatus" minOccurs="0"/>
        <xs:element ref="location" minOccurs="0"/>
        <xs:element ref="maxTimeAllowed" minOccurs="0"/>
        <xs:element ref="mode" minOccurs="0"/>
        <xs:element ref="objectives" minOccurs="0"/>
        <xs:element ref="progressMeasure" minOccurs="0"/>
        <xs:element ref="rawPassingScore" minOccurs="0"/>
        <xs:element ref="scaledPassingScore" minOccurs="0"/>
        <xs:element ref="score" minOccurs="0"/>
        <xs:element ref="sessionTime" minOccurs="0"/>
```

```
            <xs:element ref="successStatus" minOccurs="0"/>
            <xs:element ref="suspendData" minOccurs="0"/>
            <xs:element ref="timeLimitAction" minOccurs="0"/>
            <xs:element ref="totalTime" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
    <xs:element name="commentsFromLearner">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.1:    Comments from learner
        </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="commentFromLearner" type="commentType"
              minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="collectionType" fixed="array"/>
        <xs:attribute name="spm" fixed="250"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="commentsFromLMS">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.2: Comments from LMS
        </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="commentFromLMS" type="commentType"
              minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="collectionType" fixed="array"/>
        <xs:attribute name="spm" fixed="100"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="completionStatus" type="completionStatusType">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.3: Completion status
```

```
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="completionThreshold"    type="progressMeasureType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.1.4: Completion threshold
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="credit">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.1.5: Credit
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:token">
            <xs:enumeration value="credit"/>
            <xs:enumeration value="no_credit"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="dataModelVersion" type="literalString250Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.1.6: Data model version
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="entry">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.1.7: Entry
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:token">
            <xs:enumeration value="ab_initio"/>
            <xs:enumeration value="resume"/>
```

```
            <xs:enumeration value=""/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
   <xs:element name="exit">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Implements Clause 6.1.8: Exit
         </xs:documentation>
      </xs:annotation>
      <xs:simpleType>
         <xs:restriction base="xs:token">
            <xs:enumeration value="logout"/>
            <xs:enumeration value="normal"/>
            <xs:enumeration value="suspend"/>
            <xs:enumeration value="timeout"/>
            <xs:enumeration value=""/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
   <xs:element name="interactions" type="interactionsType">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Implements Clause 6.1.9: Interactions
         </xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="launchData" type="literalString4000Type">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Implements Clause 6.1.10: Launch data
         </xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="learnerId" type="longIdentifierType">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Implements Clause 6.1.11: Learner ID
         </xs:documentation>
      </xs:annotation>
```

```
</xs:element>
<xs:element name="learnerName" type="localizedString250Type">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.1.12: Learner name
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="learnerPreferenceData"
    type="learnerPreferenceType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.13: Learner preference data
      </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="lessonStatus" type="legacyStatusType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.1.14: Lesson status
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="location" type="literalString1000Type">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.1.15: Location
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="maxTimeAllowed" type="timeIntervalType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.1.16: Max time allowed
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="mode">
  <xs:annotation>
    <xs:documentation xml:lang="en">
```

```
            Implements Clause 6.1.17: Mode
          </xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="browse"/>
            <xs:enumeration value="normal"/>
            <xs:enumeration value="review"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="objectives" type="objectivesType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.18: Objectives
          </xs:documentation>
        </xs:annotation>
        <xs:unique name="uniqueInSetOfObjectives">
          <xs:selector xpath="./t:objective"/>
          <xs:field xpath="t:identifier"/>
        </xs:unique>
      </xs:element>
      <xs:element name="progressMeasure" type="progressMeasureType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.19: Progress measure
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="rawPassingScore" type="real7Type">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.20: Raw passing score
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="scaledPassingScore" type="scaledScoreType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.21: Scaled passing score
```

```
    </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="score" type="scoreType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.22: Score
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="sessionTime" type="timeIntervalType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.23: Session time
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="successStatus" type="successStatusType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.24: Success status
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="suspendData"    type="literalString4000Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.25: Suspend data
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="timeLimitAction">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Implements Clause 6.1.26: Time limit action
      </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="continue_message"/>
```

```
            <xs:enumeration value="continue_no_message"/>
            <xs:enumeration value="exit_message"/>
            <xs:enumeration value="exit_no_message"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="totalTime" type="timeIntervalType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.1.27: Total time
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- -->
<!-- Global type declarations defined by numbered clauses in
    IEEE 1484.11.1 -->
<!-- -->
<xs:complexType name="commentType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.1: Comment type
        </xs:documentation>
    </xs:annotation>
    <xs:all>
        <xs:element name="comment" type="localizedString4000Type"/>
        <xs:element name="location" type="literalString1000Type"
            minOccurs="0"/>
        <xs:element name="timeStamp" type="dateTimeType"
            minOccurs="0"/>
    </xs:all>
</xs:complexType>
<xs:simpleType name="completionStatusType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.2: Completion status
        type     </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
        <xs:enumeration value="completed"/>
        <xs:enumeration value="incomplete"/>
```

```
        <xs:enumeration value="not_attempted"/>
        <xs:enumeration value="unknown"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="dateTimeType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.3: Date time type
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:dateTime"/>
</xs:simpleType>
<xs:simpleType name="languageType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.4: Language type. Must remain simpleType
            because it is used as value for attributes
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:language">
        <xs:annotation>
            <xs:appinfo>
                <!-- <spm>250</spm> -->
            </xs:appinfo>
        </xs:annotation>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="localizedStringType" abstract="true">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.5: Localized string type.
            The Localized String Type is implemented as several
            variations with embedded SPM information.
            The SPM is not enforced by XML validators but may
            be useful for applications.
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="literalStringType">
            <xs:attribute name="lang" type="languageType"/>
```

```
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="localizedString250Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.5: Localized string type with
            SPM=250
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="localizedStringType">
            <xs:attribute name="spm" fixed="250"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="localizedString4000Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.5: Localized string type with SPM=4000
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="localizedStringType">
            <xs:attribute name="spm" fixed="4000"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="longIdentifierType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implements Clause 6.2.6: Long identifier type
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attribute name="spm" fixed="4000"/>
      </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

```
<xs:simpleType name="progressMeasureType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.2.7: Progress measure type
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="real7Type">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="scoreType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.2.8: Score type
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="scaled" type="scaledScoreType" minOccurs="0"/>
    <xs:element name="max" type="real7Type" minOccurs="0"/>
    <xs:element name="min" type="real7Type" minOccurs="0"/>
    <xs:element name="raw" type="real7Type" minOccurs="0"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="shortIdentifierType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.2.9: Short identifier type
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attribute name="spm" fixed="250"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="successStatusType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.2.10: Success status type
```

```
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
        <xs:enumeration value="failed"/>
        <xs:enumeration value="passed"/>
        <xs:enumeration value="unknown"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="real7Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            As explained in IEEE 1484.11.1-2004, Annex B.1 Real data type
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:decimal"/>
</xs:simpleType>
<xs:simpleType name="timeIntervalType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            As explained in IEEE 1484.11.1-2004, Annex B.2 Time interval
            Data type
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:duration"/>
</xs:simpleType>
<!-- -->
<!-- Above this, things follow the order of the Data Model
     document clauses. -->
<!-- ============================================================ -->
<!-- Below this are things that did not fit neatly above. They are
     organized as elements, attributes, groups, simple types, complex
     types and alphabetically by name within each of those
     categories. -->
<!-- -->
<!-- === ELEMENTS === -->
<!-- Organized in alphabetic order by element name -->
<!-- -->
<xs:element name="choices" type="setOfChoicesType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
```

```
            Set of short identifiers for interaction type "multiple
            choice" as specified in 6.1.9.5: Correct response and 6.1.9.7:
            Learner response.
        </xs:documentation>
    </xs:annotation>
    <xs:unique name="uniqueInChoicesIds">
        <xs:selector xpath="./t:choice"/>
        <xs:field xpath="."/>
    </xs:unique>
</xs:element>
<!-- -->
<!-- === ATTRIBUTES === -->
<!-- Organized in alphabetic order by attribute name -->
<!-- -->
<xs:attribute name="collectionType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            The collectionType attribute is used to inject Data Model
            information about aggregation that cannot be expressed in XML
            schema. When defined for an element or type in this schema,
            this attribute is given a fixed values. Even if the attribute
            and value are not specified in an XML instance, the XML schema
            processor makes them available to the processing application.
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:token">
            <xs:enumeration value="bag"/>
            <xs:enumeration value="array"/>
            <xs:enumeration value="set"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="spm" type="xs:integer">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            The spm attribute is used to inject Data Model information
            about SPM that cannot be expressed in XML schema. When defined
            for an element or type in this schema, this attribute is given
            a fixed values. Even if the attribute and value are not
```

```
            specified in an XML instance, the XML schema processor makes
            them available to the processing application.
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<!-- -->
<!-- === GROUPS === -->
<!-- Organized in alphabetic order by group name -->
<!-- -->
<xs:group name="grpCorrectFillIn">
    <xs:sequence>
        <xs:annotation>
            <xs:appinfo>
                <!--
                    <spm>5</spm><collectionType>bag</collectionType>
                -->
            </xs:appinfo>
        </xs:annotation>
        <xs:element name="fillMatches" minOccurs="0"
            maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="matchText" type="localizedString250Type"
                        maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="caseMatters" type="trueFalseType"
                    use="optional" default="false"/>
                <xs:attribute name="orderMatters" type="trueFalseType"
                    use="optional" default="true"/>
                <xs:attribute name="collectionType" fixed="array"/>
                <xs:attribute name="spm" fixed="10"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:group>
<xs:group name="grpCorrectLikert">
    <xs:sequence>
        <xs:element name="choice" type="shortIdentifierType"
            minOccurs="0"/>
    </xs:sequence>
```

```
</xs:group>
<xs:group name="grpCorrectLongFillIn">
  <xs:sequence>
    <xs:annotation>
      <xs:appinfo>
        <!--
          <spm>5</spm><collectionType>bag</collectionType>
        -->
      </xs:appinfo>
    </xs:annotation>
    <xs:element name="matchText" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="localizedString4000Type">
            <xs:attribute name="caseMatters" type="trueFalseType"
              use="optional" default="false"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:group>
<xs:group name="grpCorrectMatching">
  <xs:sequence>
    <xs:annotation>
      <xs:appinfo>
        <!--
          <spm>5</spm><collectionType>bag</collectionType>
        -->
      </xs:appinfo>
    </xs:annotation>
    <xs:element name="matchPattern" type="matchingPairsType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>
<xs:group name="grpCorrectMultipleChoice">
  <xs:sequence>
    <xs:annotation>
      <xs:appinfo>
        <!--
```

```
            spm>10</spm><collectionType>set</collectionType>
          -->
        </xs:appinfo>
      </xs:annotation>
      <xs:element ref="choices" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="grpCorrectNumeric">
    <xs:sequence>
      <xs:element name="min" type="real7Type" minOccurs="0"/>
      <xs:element name="max" type="real7Type" minOccurs="0"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="grpCorrectOther">
    <xs:sequence>
      <xs:element name="correctOther" type="literalString4000Type"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="grpCorrectPerformance">
    <xs:sequence>
      <xs:annotation>
        <xs:appinfo>
          <!--
            <spm>5</spm><collectionType>bag</collectionType>
          -->
        </xs:appinfo>
      </xs:annotation>
      <xs:element name="performancePattern"
          type="correctPerformancePatternType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="grpCorrectSequencing">
    <xs:sequence>
      <xs:annotation>
        <xs:appinfo>
          <!--
            <spm>5</spm><collectionType>bag</collectionType>
          -->
        </xs:appinfo>
      </xs:annotation>
```

```
        <xs:element name="stepSequence" type="stepSequenceType"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:group>
    <xs:group name="grpCorrectTrueFalse">
      <xs:sequence>
        <xs:element name="trueOrFalse" type="trueFalseType"/>
      </xs:sequence>
    </xs:group>
    <!-- variant groups for interaction responses -->
    <xs:group name="grpResponseFillIn">
      <xs:sequence>
        <xs:annotation>
          <xs:appinfo>
            <!--
              <spm>10</spm><collectionType>array</collectionType>
            -->
          </xs:appinfo>
        </xs:annotation>
        <xs:element name="fillString" type="localizedString250Type"
          minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:appinfo>
              <!-- <spm>250</spm> -->
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:group>
    <xs:group name="grpResponseLikert">
      <xs:sequence>
        <xs:element name="choice" type="shortIdentifierType"
          minOccurs="0"/>
      </xs:sequence>
    </xs:group>
    <xs:group name="grpResponseLongFillIn">
      <xs:sequence>
        <xs:element name="longFillString" type="localizedString4000Type"
          minOccurs="0"/>
      </xs:sequence>
```

```
</xs:group>
<xs:group name="grpResponseMatching">
   <xs:sequence>
      <xs:element name="matchPattern" type="matchingPairsType"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseMultipleChoice">
   <xs:sequence>
      <xs:element ref="choices"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseNumeric">
   <xs:sequence>
      <xs:element name="number" type="real7Type" minOccurs="0"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseOther">
   <xs:sequence>
      <xs:element name="responseOther" type="literalString4000Type"/>
   </xs:sequence>
</xs:group>
<!-- -->
<xs:group name="grpResponsePerformance">
   <xs:annotation>
      <xs:documentation xml:lang="en">
         The learner response for interaction type "performance"
         as specified in 6.1.9.7: Learner response
      </xs:documentation>
   </xs:annotation>
   <xs:sequence>
      <xs:annotation>
         <xs:appinfo>
            <!--
               <spm>250</spm><collectionType>array</collectionType>
            -->
         </xs:appinfo>
      </xs:annotation>
      <xs:element name="step" type="learnerPerformanceStepType"
         minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
```

```
</xs:group>
<xs:group name="grpResponseSequencing">
   <xs:sequence>
      <xs:element name="steps" type="stepSequenceType" minOccurs="0"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseTrueFalse">
   <xs:sequence>
      <xs:element name="trueOrFalse" type="trueFalseType"
         minOccurs="0"/>
   </xs:sequence>
</xs:group>
<!-- -->
<!-- === SIMPLE TYPES === -->
<!-- Organized in alphabetic order by type name -->
<!-- -->
<xs:simpleType name="interactionResultType">
   <xs:annotation>
      <xs:documentation xml:lang="en">
         Reusable type definition used for 6.1.9.8: Result
         The value of result can be either a numeric value
         or a specified token. This element uses xs:union to avoid
         having to define sub-elements with arbitrary names.
      </xs:documentation>
   </xs:annotation>
   <xs:union memberTypes="real7Type interactionResultTokenType"/>
</xs:simpleType>
<xs:simpleType name="interactionResultTokenType">
   <xs:restriction base="xs:token">
      <xs:enumeration value="correct"/>
      <xs:enumeration value="incorrect"/>
      <xs:enumeration value="neutral"/>
      <xs:enumeration value="unanticipated"/>
   </xs:restriction>
</xs:simpleType>
<xs:simpleType name="interactionTypeType">
   <xs:annotation>
      <xs:documentation xml:lang="en">
         Implements Clause 6.1.9.2: Type
      </xs:documentation>
```

```
            </xs:annotation>
                <xs:restriction base="xs:token">
                   <xs:enumeration value="true_false"/>
                   <xs:enumeration value="multiple_choice"/>
                   <xs:enumeration value="fill_in"/>
                   <xs:enumeration value="long_fill_in"/>
                   <xs:enumeration value="likert"/>
                   <xs:enumeration value="matching"/>
                   <xs:enumeration value="performance"/>
                   <xs:enumeration value="sequencing"/>
                   <xs:enumeration value="numeric"/>
                   <xs:enumeration value="other"/>
                </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="literalStringType">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Ensures preservation of whitespace
         </xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string">
         <xs:whiteSpace value="preserve"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="legacyStatusType">
      <xs:restriction base="xs:token">
         <xs:enumeration value="browsed"/>
         <xs:enumeration value="completed"/>
         <xs:enumeration value="failed"/>
         <xs:enumeration value="incomplete"/>
         <xs:enumeration value="not_attempted"/>
         <xs:enumeration value="passed"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="scaledScoreType">
      <xs:restriction base="real7Type">
         <xs:minInclusive value="-1"/>
         <xs:maxInclusive value="1"/>
      </xs:restriction>
   </xs:simpleType>
```

```
<xs:simpleType name="trueFalseType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      True and false options for interaction type "true_false"
      as specified in 6.1.9.5: Correct response and 6.1.9.7:
      Learner response.
      Also used for tokens for various other boolean elements.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- === COMPLEX TYPES === -->
<!-- Organized in alphabetic order by type name -->
<!-- -->
<xs:complexType name="setOfChoicesType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Set of short identifiers for interaction type "multiple choice"
      as specified in 6.1.9.5: Correct response and 6.1.9.7: Learner
      response.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="choice" type="shortIdentifierType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="collectionType" fixed="set"/>
  <xs:attribute name="spm" fixed="36"/>
</xs:complexType>
<xs:complexType name="correctPerformancePatternType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A correct response record (order_matters + answers)
      For interaction type "performance"
      as specified in 6.1.9.5: Correct response
    </xs:documentation>
```

```
    </xs:annotation>
  <xs:sequence>
    <xs:element name="step" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:all>
          <xs:element name="stepName" type="shortIdentifierType"
            minOccurs="0"/>
          <xs:element name="stepAnswer" minOccurs="0">
            <xs:complexType>
              <xs:choice>
                <xs:element name="literal"
                  type="literalString250Type" minOccurs="0"/>
                <xs:element name="numeric" minOccurs="0">
                  <xs:complexType>
                    <xs:attribute name="min" type="real7Type"/>
                    <xs:attribute name="max" type="real7Type"/>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="orderMatters" type="trueFalseType"
    use="optional" default="true"/>
  <xs:attribute name="collectionType" fixed="array"/>
  <xs:attribute name="spm" fixed="250"/>
</xs:complexType>
<xs:complexType
  name="correctResponsesType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implements Clause 6.1.9.5: Correct responses.
      Note: It is up to the implementation to choose the correct
      group to match the interaction type. This correspondence cannot
      be expressed or validated using XML Schema.
    </xs:documentation>
  </xs:annotation>
```

```
    <xs:choice>
      <xs:group ref="grpCorrectTrueFalse"/>
      <xs:group ref="grpCorrectMultipleChoice"/>
      <xs:group ref="grpCorrectFillIn"/>
      <xs:group ref="grpCorrectLongFillIn"/>
      <xs:group ref="grpCorrectLikert"/>
      <xs:group ref="grpCorrectMatching"/>
      <xs:group ref="grpCorrectPerformance"/>
      <xs:group ref="grpCorrectSequencing"/>
      <xs:group ref="grpCorrectNumeric"/>
      <xs:group ref="grpCorrectOther"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="interactionType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Reusable type to implements a single interaction record as
        defined in Clause 6.1.9.
      </xs:documentation>
    </xs:annotation>
    <xs:all>
      <xs:element name="identifier" type="longIdentifierType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.9.1: ID
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="type" type="interactionTypeType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.9.2: Type
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="objectiveIds" type="objectiveIdsType"
        minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            Implements Clause 6.1.9.3: Objectives ID
```

```
        </xs:documentation>
      </xs:annotation>
      <xs:unique name="uniqueInObjectivesIds">
        <xs:selector xpath="./t:objectiveId"/>
        <xs:field xpath="."/>
      </xs:unique>
    </xs:element>
    <xs:element name="timeStamp" type="dateTimeType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.9.4: Time stamp
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="correctResponses" type="correctResponsesType"
      minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.9.5: Correct responses
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="weighting" type="real7Type" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.9.6: Weighting
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="learnerResponse" type="learnerResponseType"
      minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.9.7: Learner response
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="result" type="interactionResultType"
      minOccurs="0">
      <xs:annotation>
```

```
            <xs:documentation xml:lang="en">
              Implements Clause 6.1.9.8: Result
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="latency" type="timeIntervalType" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">
              Implements Clause 6.1.9.9: Latency
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="description" type="localizedString250Type"
          minOccurs="0"/>
      </xs:all>
    </xs:complexType>
    <xs:complexType name="interactionsType">
      <xs:sequence>
        <xs:element name="interaction" type="interactionType"
          minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="collectionType" fixed="bag"/>
        <xs:attribute name="spm" fixed="250"/>
    </xs:complexType>
    <xs:complexType name="learnerPerformanceStepType">
      <xs:all>
        <xs:element name="stepName" type="shortIdentifierType"
          minOccurs="0"/>
        <xs:element name="stepAnswer" minOccurs="0">
          <xs:complexType>
            <xs:choice>
              <xs:element name="literal" type="literalString250Type"
                minOccurs="0"/>
              <xs:element name="numeric" type="real7Type"
                minOccurs="0"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
```

```
<xs:complexType name="learnerPreferenceType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Implementslearner_preference_type in Clause 6.1.13:
      Learner preference data
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="audioLevel" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.13.1: Audio level
        </xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="real7Type">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="language" type="languageType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.13.2: Language
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="deliverySpeed" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Implements Clause 6.1.13.3: Delivery speed
        </xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="real7Type">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="audioCaptioning" minOccurs="0">
```

```
            <xs:annotation>
               <xs:documentation xml:lang="en">
                  Implements Clause 6.1.13.4: Audio captioning
               </xs:documentation>
            </xs:annotation>
            <xs:simpleType>
               <xs:restriction base="xs:token">
                  <xs:enumeration value="off"/>
                  <xs:enumeration value="no_change"/>
                  <xs:enumeration value="on"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
      </xs:all>
   </xs:complexType>
   <xs:complexType name="learnerResponseType">
      <xs:annotation>
         <xs:documentation xml:lang="en">
            Implements Clause 6.1.9.7: Learner response.
            Note: It is up to the implementation to choose the correct
            group or element name to match the interaction type. This
            correspondence cannot be expressed or validated using
            XML  Schema.
         </xs:documentation>
      </xs:annotation>
      <xs:choice>
         <xs:group ref="grpResponseTrueFalse"/>
         <xs:group ref="grpResponseMultipleChoice"/>
         <xs:group ref="grpResponseFillIn"/>
         <xs:group ref="grpResponseLongFillIn"/>
         <xs:group ref="grpResponseLikert"/>
         <xs:group ref="grpResponseMatching"/>
         <xs:group ref="grpResponsePerformance"/>
         <xs:group ref="grpResponseSequencing"/>
         <xs:group ref="grpResponseNumeric"/>
         <xs:group ref="grpResponseOther"/>
      </xs:choice>
   </xs:complexType>
   <xs:complexType name="literalString250Type">
      <xs:annotation>
```

```
        <xs:documentation xml:lang="en">
            Implement any literal string with SPM=250
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="literalStringType">
            <xs:attribute name="spm" fixed="250"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="literalString1000Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implement any literal string with SPM=1000
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="literalStringType">
            <xs:attribute name="spm" fixed="1000"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="literalString4000Type">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Implement any literal string with SPM=4000
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="literalStringType">
            <xs:attribute name="spm" fixed="4000"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="matchingPairType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            A pair of matched short identifiers for interaction type
            "matching" as specified in 6.1.9.5: Correct response and
            6.1.9.7: Learner response.
```

```
        </xs:documentation>
    </xs:annotation>
    <xs:all>
        <xs:element name="source" type="shortIdentifierType"/>
        <xs:element name="target" type="shortIdentifierType"/>
    </xs:all>
</xs:complexType>
<xs:complexType name="matchingPairsType">
    <xs:sequence>
        <xs:annotation>
            <xs:documentation xml:lang="en">
                A collection of 0 or more instances of matchingPairType.
            </xs:documentation>
        </xs:annotation>
        <xs:element name="pair" type="matchingPairType" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="collectionType" fixed="bag"/>
    <xs:attribute name="spm" fixed="36"/>
</xs:complexType>
<xs:complexType name="objectiveIdsType">
    <xs:sequence>
        <xs:element name="objectiveId" type="longIdentifierType"
            minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="collectionType" fixed="array"/>
    <xs:attribute name="spm" fixed="10"/>
</xs:complexType>
<xs:complexType
    name="objectivesType">
    <xs:sequence>
        <xs:element name="objective" type="objectiveType" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="collectionType" fixed="set"/>
    <xs:attribute name="spm" fixed="100"/>
</xs:complexType>
<xs:complexType name="objectiveType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
```

```
            Reusable type to implement the objective_type record
            defined in Clause 6.1.18
          </xs:documentation>
        </xs:annotation>
        <xs:all>
          <xs:element name="identifier" type="longIdentifierType"/>
          <xs:element name="score" type="scoreType" minOccurs="0"/>
          <xs:element name="status" type="legacyStatusType" minOccurs="0"/>
          <xs:element name="progressMeasure" type="progressMeasureType"
            minOccurs="0"/>
          <xs:element name="completionStatus"
            type="completionStatusType"    minOccurs="0"/>
          <xs:element name="successStatus" type="successStatusType"
            minOccurs="0"/>
          <xs:element name="description" type="localizedString250Type"
            minOccurs="0"/>
        </xs:all>
      </xs:complexType>
      <xs:complexType name="responseOtherType">
        <xs:simpleContent>
          <xs:extension base="literalString4000Type"/>
        </xs:simpleContent>
      </xs:complexType>
      <xs:complexType name="stepSequenceType">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            A sequence of steps for interaction type "sequencing"
            as specified in 6.1.9.5: Correct response and 6.1.9.7:
            Learner response.
            The same identifier may appear more than once in the list,
            because a step may be repeated in an interaction.
.         </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element name="step" type="shortIdentifierType" minOccurs="0"
            maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="collectionType" fixed="array"/>
          <xs:attribute name="spm" fixed="36"/>
      </xs:complexType>
```

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

附錄 C

(參考)

COCD XML 實例範例

圖 C.1 顯示 COCD XML 實例的範例，此範例舉例說明了附錄 B 中 XSD 所定義的全部元件及屬性。

Annex C

(informative)

An example COCD XML instance

Figure C.1 shows a COCD XML instance that instantiates all elements and attributes defined in the XSD in Annex B.

圖C.1 COCD XML實例範例

Figure C.1—An example COCD XML instance

```
<?xml version="1.0" encoding="UTF-8"?>
<cocd xmlns="http://ltsc.ieee.org/xsd/1484_11_3"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ltsc.ieee.org/xsd/1484_11_3
      ieee_1484_11_3_2005.xsd">
   <commentsFromLearner>
      <commentFromLearner>
         <comment lang="en-us">Having a good time</comment>
         <location>Somewhere</location>
         <timeStamp>2005-10-17T09:30:47-05:00</timeStamp>
      </commentFromLearner>
   </commentsFromLearner>
   <commentsFromLMS>
      <commentFromLMS>
         <comment lang="en-us">Have a good time</comment>
         <location>Wherever it may be</location>
         <timeStamp>2005-10-17T08:30:02-05:00</timeStamp>
      </commentFromLMS>
   </commentsFromLMS>
   <completionStatus>completed</completionStatus>
   <completionThreshold>0.9</completionThreshold>
   <credit>credit</credit>
   <dataModelVersion>1484.11.1</dataModelVersion>
   <entry>ab_initio</entry>
   <exit>logout</exit>
```

```
<launchData>Nothing special</launchData>
<learnerId>1234-foobar-1234</learnerId>
<learnerName lang="en-us">John Doe</learnerName>
<learnerPreferenceData>
   <audioLevel>1</audioLevel>
   <language>en-us</language>
   <deliverySpeed>1.25</deliverySpeed>
   <audioCaptioning>off</audioCaptioning>
</learnerPreferenceData>
<lessonStatus>browsed</lessonStatus>
<location>Page 4, paragraph 6</location>
<maxTimeAllowed>P2M3DT10H30M</maxTimeAllowed>
<mode>browse</mode>
<interactions>
   <interaction>
      <identifier>urn:ostyn.com:TFQ2003090412345</identifier>
      <objectiveIds>
         <objectiveId>1234-foobar-1234</objectiveId>
         <objectiveId>1235-barfoo-1345</objectiveId>
      </objectiveIds>
       <timeStamp>2005-10-17T09:30:17-05:00</timeStamp>
       <description>Vanilla is lighter than chocolate. True or
      false?</description>
        <result>unanticipated</result>
       <latency>PT23.4S</latency>
       <weighting>.33</weighting>
       <type>true_false</type>
       <correctResponses>
          <trueOrFalse>true</trueOrFalse>
       </correctResponses>
       <learnerResponse/>
   </interaction>
<interaction>
   <identifier>urn:ostyn.com:MCQ2003090412345</identifier>
   <objectiveIds>
      <objectiveId>
      1234-foobar-1234</objectiveId>
      <objectiveId>1235-barfoo-1345
    </objectiveId>
   </objectiveIds>
```

```
    <timeStamp>2005-10-17T09:32:47.45-05:00</timeStamp>
    <description>What was J.H.'s favorite ice cream
    combination?</description>
    <result>incorrect</result>
    <latency>PT23.4S</latency>
    <weighting>.33</weighting>
    <type>multiple_choice</type>
    <correctResponses>
      <choices>
        <choice>Vanilla</choice>
      </choices>
      <choices>
        <choice>Vanilla</choice>
      </choices>
      <choices>
        <choice>Chocolate</choice>
      </choices>
      <choices>
        <choice>Chocolate</choice>
        <choice>Vanilla</choice>
      </choices>
    </correctResponses>
    <learnerResponse>
      <choices>
        <choice>Pistachio</choice>
        <choice>Chocolate</choice>
      </choices>
    </learnerResponse>
  </interaction>
  <interaction>

    <identifier>urn:ostyn.com:FIQ2003090412345</identifier>
    <objectiveIds>
      <objectiveId>1234-foobar-1234</objectiveId>
      <objectiveId>1235-barfoo-1345</objectiveId>
    </objectiveIds>
    <timeStamp>2005-10-17T09:34:47-05:00</timeStamp>
    <description>Match things with numbers</description>
    <result>3.5926</result>
    <latency>PT30M</latency>
```

```
      <weighting>3</weighting>
      <type>fill_in</type>
       <correctResponses>
        <fillMatches caseMatters="false" orderMatters="false">
          <matchText lang="en">some</matchText>
          <matchText lang="en">thing</matchText>
         </fillMatches>
        <fillMatches caseMatters="false" orderMatters="true">
          <matchText>1</matchText>
          <matchText>2</matchText>
        </fillMatches>
      </correctResponses>
      <learnerResponse>
         <fillString lang="en">This is just any short response.</fillString>
      </learnerResponse>
   </interaction>
   <interaction>
      <identifier>urn:ostyn.com:LFQ2003090412345</identifier>
      <objectiveIds>
        <objectiveId>
        1234-foobar-1234</objectiveId>
        <objectiveId>
        1235-barfoo-1345
      </objectiveId>
      </objectiveIds>
      <timeStamp>2005-10-17T09:36:47-05:00</timeStamp>
      <description lang="fr">Début de la Ballade des Pendus</description>
      <result>-0.1415926</result>
      <latency>PT23.4S</latency>
      <weighting>33</weighting>
      <type>long_fill_in</type>
      <correctResponses>
        <matchText lang="fr" caseMatters="false">Frères humains qui après nous
vivez, N'ayez les coeurs contre nous endurcis</matchText>
        <matchText lang="fr" caseMatters="false">Frères humains qui après nous
vivez, N'ayez les cuers contre nous endurcis</matchText>
         <matchText lang="fr" caseMatters="true">Frères humains qui après nous
vivez</matchText>
      </correctResponses>
      <learnerResponse>
```

```
        <longFillString lang="fr-BE">Ça commence avec "Frères
humains..."</longFillString>
      </learnerResponse>
    </interaction>
    <interaction>
      <identifier>urn:ostyn.com:LIQ2003090412345</identifier>
      <objectiveIds>
        <objectiveId>
        1234-foobar-1234
      </objectiveId>
      </objectiveIds>
      <timeStamp>2005-10-17T09:38:47-05:00</timeStamp>
      <description>Which approach is most likely to succeed?</description>
      <latency>PT23.4S</latency>
      <weighting>.33</weighting>
      <type>likert</type>
      <learnerResponse>
        <choice>option_5</choice>
      </learnerResponse>
    </interaction>
    <interaction>
      <identifier>urn:ostyn.com:MAQ2003090412345</identifier>
      <objectiveIds>
        <objectiveId>
        1234-foobar-1234</objectiveId>
        <objectiveId>
        1235-barfoo-1345
      </objectiveId>
      </objectiveIds>
      <timeStamp>2005-10-17T09:40:47-05:00</timeStamp>
      <description>Connect the shmiblicks to the corresponding
      garfubles</description>
      <result>incorrect</result>
      <latency>PT23.4S</latency>
      <weighting>.33</weighting>
      <type>matching</type>
      <correctResponses>
        <matchPattern>
          <pair>
            <source>something_A</source>
```

```
            <target>something_B</target>
          </pair>
          <pair>
            <source>something_C</source>
            <target>something_D</target>
          </pair>
          <pair>
            <source>something_E</source>
            <target>something_F</target>
          </pair>
        </matchPattern>
        <matchPattern>
          <pair>
            <source>something_C</source>
            <target>something_D</target>
          </pair>
          <pair>
            <source>something_E</source>
            <target>something_F</target>
          </pair>
        </matchPattern>
      </correctResponses>
      <learnerResponse>
        <matchPattern>
          <pair>
            <source>something_C</source>
            <target>something_D</target>
          </pair>
          <pair>
            <source>something_E</source>
            <target>something_F</target>
          </pair>
        </matchPattern>
      </learnerResponse>
    </interaction>
    <interaction>
      <identifier>urn:ostyn.com:PEQ2003090412345</identifier>
      <objectiveIds>
        <objectiveId>
        1234-foobar-1234</objectiveId>
```

```
    <objectiveId>
    1235-barfoo-1345
</objectiveId>
</objectiveIds>
<timeStamp>2005-10-17T09:42:47-05:00</timeStamp>
<description>Steps to diagnose the schmiblick</description>
<result>3.5</result>
<latency>PT23.4S</latency>
<weighting>.33</weighting>
<type>performance</type>
<correctResponses>
   <performancePattern orderMatters="true">
      <step>
         <stepName>StepB</stepName>
         <stepAnswer>
            <numeric min="10" max="10"/>
         </stepAnswer>
      </step>
      <step>
         <stepName>StepC</stepName>
         <stepAnswer>
            <literal>Green whatchamakalit</literal>
         </stepAnswer>
      </step>
      <step>
          <stepName>stepD</stepName>
      </step>
   </performancePattern>
   <performancePattern orderMatters="false">
      <step>
         <stepName>stepA</stepName>
         <stepAnswer>
            <literal>Push the diagnostic button</literal>
         </stepAnswer>
      </step>
      <step>
         <stepName>stepE</stepName>
      </step>
   </performancePattern>
</correctResponses>
```

```
    <learnerResponse>
      <step>
        <stepName>StepC</stepName>
        <stepAnswer>
          <literal>Blue whatchamakalit</literal>
        </stepAnswer>
      </step>
      <step>
        <stepName>StepD</stepName>
        <stepAnswer>
          <numeric>8.7</numeric>
        </stepAnswer>
      </step>
      <step>
        <stepName>StepD</stepName>
      </step>
    </learnerResponse>
  </interaction>
  <interaction>
    <identifier>urn:ostyn.com:INT2003090412345</identifier>
    <objectiveIds>
      <objectiveId>
      urn:bar.com/RCD/2345-800df-4%20test</objectiveId>
      <objectiveId>
      machinchose1234
     </objectiveId>
    </objectiveIds>
    <timeStamp>2005-10-17T09:44:47-05:00</timeStamp>
    <description>Steps to buy and enjoy ice cream.</description>
    <result>3.14159</result>
    <latency>PT23.4S</latency>
    <weighting>.33</weighting>
    <type>sequencing</type>
    <correctResponses>
      <stepSequence>
        <step>Choose_flavor</step>
        <step>Order_ice_cream</step>
        <step>Eat_ice_cream</step>
        <step>Wipe_chin</step>
       </stepSequence>
```

```
        <stepSequence>
          <step>Raid_fridge</step>
          <step>Choose_flavor</step>
          <step>Eat_ice_cream</step>
          <step>Wipe_chin</step>
        </stepSequence>
    </correctResponses>
    <learnerResponse>
      <steps>
        <step>Order_ice_cream</step>
        <step>Choose_flavor</step>
        <step>Eat_ice_cream</step>
        <step>Wipe_chin</step>
        <step>Eat_ice_cream</step>
        <step>Wipe_chin</step>
      </steps>
    </learnerResponse>
</interaction>
<interaction>
    <identifier>urn:ostyn.com:NUQ2003090412345</identifier>
    <objectiveIds>
      <objectiveId>
      1234-foobar-1234
    </objectiveId>
    </objectiveIds>
    <timeStamp>2005-10-17T09:17:47-05:00</timeStamp>
    <description>Pick a likely number for the result of this
    operation.</description>
    <result>3.14159</result>
    <latency>PT23.4S</latency>
    <weighting>7</weighting>
    <type>numeric</type>
    <correctResponses>
      <min>0</min>
      <max>123456783453.1415926</max>
    </correctResponses>
    <learnerResponse>
      <number>3.1415926</number>
    </learnerResponse>
</interaction>
```

```
<interaction>
  <identifier>urn:ostyn.com:OTQ2003090412345</identifier>
  <objectiveIds>
     <objectiveId>
    1234-foobar-1234
  </objectiveId>
  </objectiveIds>
  <timeStamp>2005-10-17T09:13:47-05:00</timeStamp>
  <description>Some other kind of interaction</description>
  <result>correct</result>
  <latency>PT23.4S</latency>
  <weighting>.33</weighting>
  <type>other</type>
  <correctResponses>
    <correctOther>Something and <![CDATA[<Something>more or less
   complicated</Something>]]></correctOther>
    </correctResponses>
    <learnerResponse>
      <responseOther>Something more or less
       complicated</responseOther>
      </learnerResponse>
  </interaction>
</interactions>
<objectives>
  <objective>
    <identifier>urn:ostyn.com:id200309041234578</identifier>
    <completionStatus>incomplete</completionStatus>
    <description>Answer 10 questions</description>
    <score>
      <scaled>1.0</scaled>
      <max>77</max>
      <min>0</min>
      <raw>77</raw>
    </score>
    <status>browsed</status>
    <successStatus>failed</successStatus>
  </objective>
  <objective>
    <identifier>urn:ostyn.com:id200309041234576</identifier>
    <completionStatus>completed</completionStatus>
```

```
      </objective>
      </objectives>
    <progressMeasure>0.95</progressMeasure>
    <rawPassingScore>600</rawPassingScore>
    <scaledPassingScore>0.5</scaledPassingScore>
    <score>
       <scaled>0.5</scaled>
       <max>800</max>
       <min>400</min>
       <raw>600</raw>
    </score>
    <sessionTime>P3DT10H30M</sessionTime>
    <successStatus>passed</successStatus>
    <suspendData>Something=4; line break here:
save this white space (10 spaces)[ ]end of line
beginning of line. Something else on third line.</suspendData>
     <timeLimitAction>continue_message</timeLimitAction>
     <totalTime>P5DT10H30M</totalTime>
</cocd>
```

附錄 D

(參考)

XSD 註釋

D.1 一般(General)

本附錄乃針對附錄 B 中之 XSD 之說明與使用指南。本附錄亦須參照 IEEE 1484.11.1-2004 標準的第 6 小節。使用本附錄前須對 IEEE 1484.11.1-2004 標準有所了解。

本附錄結構將採與 IEEE 1484.11.1-2004 標準第 6 小節相同之結構。目的是能幫助同時閱讀 IEEE 1484.11.1-2004 標準與本附錄中之 XSD(見附錄 B)及 XML 實例的範例(見附錄 C)。

備考 1.本附錄不打算對每個在 XSD 中實作的設計或語法選擇提供說明或基本理由，也不打算爲 XML 或 XML 架構定義語言做個別指導。了解本附錄前建議熟悉 W3C XML 架構之第一部份和第二部分。本附錄提及由 W3C XML 架構定義語言具體指明的特點和對現行 XML 架構處理器實作的測試。

2.本附錄所有範例的 XML 中，xs:prefix 代表型式，元件，群組，或由 W3C 名稱領域〝http://www.w3.org/2001/XMLSchema〞裡定義的屬性名稱。

3.本附錄前半段的 XSD 乃依據 IEEE 1484.11.1-2004 標準第 6 小節的次序撰寫。後半段之 XSD 則使用慣用的群組元件和型式宣告：首先是元件然後是屬性，群組，簡單型式，和複雜型式。每一群組範圍內，元件皆按照名稱字母次序排列。

Annex D

(informative)

Explanatory XSD notes

D.1 General

This annex is a guide to the understanding and use of the XSD in Annex B. This annex references specific subclauses of Clause 6 of IEEE Std 1484.11.1-2004. IEEE Std 1484.11.1-2004 is required to understand this annex.

Where possible, this annex is organized in the same order as the subclauses of Clause 6 of IEEE Std 1484.11.1-2004. The intent is to facilitate a parallel reading of this annex with IEEE Std 1484.11.1-2004, the XSD (see Annex B), and the sample XML instance (see Annex C).

NOTE 1—This annex is not intended to provide an explanation or rationale for every design or syntax choice implemented in the XSD nor is it intended to be an XML or XML Schema definition language tutorial. Familiarity with W3C XML Schema, Parts 1 and 2, is required to understand this annex. This annex mentions features specified by the W3C XML Schema definition language and tested with current implementations of XML schema processors.

NOTE 2—In all examples, XML fragments in this annex, the xs: prefix denotes a type, element, group,

or attribute name defined by the W3C namespace “http://www.w3.org/2001/XMLSchema”.

NOTE 3—The first half of the XSD follows the order of the subclauses of Clause 6 of IEEE Std 1484.11.1-2004. The second half of the XSD uses conventional groupings of elements and type declarations: first elements and then attributes, groups, simple types, and complex types. Within each grouping, the elements are arranged alphabetically by name.

D.2 W3C XML架構定義語言於資料模型呈現之限制(Limitations of the W3C XML Schema definition language for the representation of the data model)

在W3C XML 架構定義語言中，部分定義於資料模型中的需求無法被表達，如藉由 XML 架構處理程式自動驗證或限制的實作。這些需求將於以下D.2.1 到 D.2.4 中討論。

D.2 Limitations of the W3C XML Schema definition language for the representation of the data model

Some requirements defined in the data model cannot be expressed in the W3C XML Schema definition language in any way that would allow automatic validation or constraint enforcement by a generic XML Schema processor. These requirements are discussed in D.2.1 through D.2.4.

D.2.1 紀錄袋、陣列與集合之編碼(Encoding of bags, arrays and sets)

由資料模型定義的紀錄袋(bags)、陣列和集合是以 XML 架構組合器 xs：sequence 實作。對於類似元件的蒐集則使用 xs：sequence，因爲此係唯一能同時包含多個元件的方法，即使 xs：sequence 隱含複雜型式的教材必須看似一次序列表。下述範例爲一典型紀錄袋/陣列/集合構造於 XSD 中。

```
<xs:element name="objectives">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="objective" type="objectiveType"
        minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

D.2.1 Encoding of bags, arrays and sets

The bags, arrays, and sets defined by the data model are implemented with the XML Schema compositor xs:sequence. For collections of like elements, xs:sequence is used because it is the only way to allow a cardinality of more than one for the contained element, even though xs:sequence implies that the contents of the complex type have to appear as an ordered list. A typical XSD construct for a bag, array, or set is similar to the following example.

此範例這可能像下述的範例一樣在 COCD XML 實例中被舉明。

```
<objectives>
        <objective>
            <identifier>urn:foo.com:id200309041234578</identifier>
            <completionStatus>incomplete</completionStatus>
        </objective>
        <objective>
            <identifier>urn:foo.com:id200309041234534</identifier>
            <completionStatus>completed</completionStatus>
        </objective>
    </objectives>
```

This may be instantiated in a COCD XML instance as in the following example.

W3C XML 架構定義語言不能具體指明是否與 xs:sequence 一起實作的集合應被看作成紀錄袋(無序的)或陣列(次序的)。然而，在某些案例中有可能具體指明是否集合中的項目必須擁有不同的教材，其爲集合模型的特性。實作者應該參照 IEEE 1484.11.1-2004 標準來決定是否定義在序列中有單一元件被定義成 xs：sequence 的特定元件，應該被看作是一個紀錄袋，一個集合，或一個陣列。按照 W3C XML 第二部分的規定，〝此規格不爲一些資料型式定義一次序關係的事實不表示一些其他的應用不能看待資料型式是利用自己次序關係而排序的。〞

The W3C XML Schema definition language cannot specify whether a collection implemented with xs:sequence should be treated as a bag (unordered) or an array (ordered). However, it is possible in some cases to specify whether the items in the collection have to have different content, which is one characteristic of a set model. Implementers should refer to IEEE Std 1484.11.1-2004 to determine whether a particular element defined as xs:sequence with a single element defined in the sequence should be treated as a bag, a set, or an array. As stated in W3C XML Schema Part 2, “The fact that this specification does not define an order-relation for some datatype does not mean that some other application cannot treat that datatype as being ordered by imposing its own order relation.”

無論什麼時候 xs：sequence 都被定義成型式定義的一部份，XSD 定義一個屬性名爲 collectionType。此屬性的值是固定的如同一個紀錄袋，陣列或集合。XML 架構語法不允許此屬性對定義在一個 xs:group 結構之元件序列的增加。collectionType 屬性對 XML 確認沒有效果，但是它可能對應用有用，因爲一個 XML 架構處理器必須產生固定的屬性和值，並且要是應用能利用的，即使它們沒有明確的在 COCD XML 實例中具體指明。本標準沒有定義任何關於使用或解釋 collectionType 屬性相符合的需求。

一個實作可能使用固定的collectionType屬性如下：

(1)舉例說明一份 XML 文件包含一個元件是給那些定義在 XML 架構處理器環境的

XSD 中的 collectionType 屬性。

(2)透過呼叫平常的 XML 架構處理器方法可以獲得 collectionType 屬性和它的值。例如，使用 XPath 表達，例如〝./@ collectionType〞，將爲現行的元件存取屬性 collectionType。此值可以在之後用來決定是否看待元件的序列爲次序的或無序的。

Whenever an xs:sequence construct is defined as part of a type definition, the XSD defines an attribute named collectionType. The value of this attribute is fixed as one of bag, array, or set. XML Schema syntax does not allow the addition of this attribute for element sequences defined in an xs:group construct. The collectionType attribute has no effect on XML validation, but it may be useful for applications, because an XML Schema processor has to make the fixed attribute and its value available to applications even if they are not explicitly specified in the COCD XML instance. This Standard does not define any conformance requirements regarding the use or interpretation of the collectionType attribute.

An implementation might use the fixed collectionType attribute as follows:

— Instantiate an XML document that contains an element for which the collectionType attribute is defined in the XSD in an XML Schema processor environment.

— Get the collectionType attribute and its value by calling usual XML Schema processor methods. For example, using an XPath expression, such as “./@collectionType”, will access the attribute collectionType for the current element. The value can then be used to determine whether to treat the sequence of elements as ordered or unordered.

D.2.2 獨特性

XML架構元件xs:unique盡可能的被用在XSD中來實作蒐集中的獨特性，其爲定義爲資料模型中的集合或用其他方法要求獨特性。這些集合如下：

(1)6.1.9 互動：每個互動需要一個識別符，此識別符必須在教材物件的情境脈絡範圍內是獨一無二的。

(2)6.1.9.3 目標 ID 在一個互動紀錄中。

(3)選擇集合的選擇在 6.1.9.5 正確回覆和 6.1.9.7 學習者對多重選擇互動回覆中。

(4)6.1.18 目標：每個目標需要一個識別符，此識別符必須在教材物件的情境脈絡範圍內是獨一無二的。

D.2.2 Uniqueness

Where possible, the standard XML Schema element xs:unique is used in the XSD to enforce uniqueness in collections that are defined as sets in the data model or that otherwise require uniqueness. These sets are as follows:

— 6.1.9 Interactions: Each interaction requires an identifier, and the identifier has to be unique within the context of the content object.

— 6.1.9.3 Objectives ID in an interaction record.
— The choices in a set of choices in 6.1.9.5 Correct responses and 6.1.9.7 Learner response for multiple choice interactions.
— 6.1.18 Objectives: Each objective requires an identifier, and the identifier has to be unique within the context of the content object.

使用 xs：unique 允許 XML 架構處理器來自動實作獨特性限制。然而，對於其他定義在資料模型中之集合，獨特性的驗證不能被自動實作。W3C XML 架構定義語言中的限制並不總是能具體指明。例如，對互動型式多重_選擇集合中正確回覆之選擇集合中，對內部集合的獨特性能夠具體指明，但對外部集合就無法具體指明。賦予外部集合限制將需要增加相當多的複雜性，例如對外部集合元件額外的任意識別符，但這依舊不能保證獨特性，因爲元件實際的內容不能被檢查。外部集合的元件將保證有獨一無二的識別符，但是不能保證有獨一無二的內容。

Using xs:unique allows an XML Schema processor to enforce the uniqueness constraint automatically. However, validation of uniqueness for the other sets defined in the data model cannot be enforced automatically. It is not always possible to specify this constraint in the W3C XML Schema definition language. For example, in the set of set of choices for the correct response for the interaction type multiple_choice, uniqueness can be specified for the inner set but not for the outer set. Enabling the constraint for the outer set would require adding considerable complexity, such as additional arbitrary identifiers for the elements of the outer set, and it still would not guarantee uniqueness, because the actual content of the elements cannot be inspected. The outer set elements would be guaranteed to have unique identifiers but would not be guaranteed to have unique content.

D.2.3 最小允許上限值(Smallest permitted maximums)

SPM 不能夠在 W3C XML 架構定義語言中用任何允許自動驗證如同定義在資料模型中 SPM 限制的方法來表達。因此 XSD 總是具體指明 maxOccur=〝無限的〞，當資料模型允許多樣性，及不對其他具體指明在資料模型中 SPM 型式設置 maxLength 屬性。實作將參照 IEEE 1484.11.1-2004 標準來決定適用特定元件的 SPM。

XSD 盡可能定義一屬性名爲 spm。此屬性被加給數個元件和型式，這些元件和型式是定義在 XSD 中且帶有一固定值，那就是 SPM 值。然而，XML 架構語法不允許此屬性對定義在 xs：group 結構元件序列的增加，也不允許此屬性爲必須保留簡單型式的字串型式定義，例如短及長識別符。此屬性對 XML 驗證沒有效果，但是它可能對應用有用，因爲 XML 架構處理器必須產生固定的屬性和值，並且要是應用能利用的，即使它們沒有明確的在 COCD XML 實例中被具體指明。本標準沒有定義任何關於使用或解釋 spm 屬性相符合的需求。

D.2.3 Smallest permitted maximums

An SPM cannot be expressed in the W3C XML Schema definition language in any way that would allow automatic validation of an SPM constraint as defined in the data model. Therefore, the XSD always specifies maxOccur="unbounded" when the data model allows multiplicity, and it does not set a maxLength attribute for other types for which an SPM is specified in the data model. Implementers should refer to IEEE Std 1484.11.1-2004 to determine the SPM that applies to a particular element.

Where possible, the XSD defines an attribute named spm. This attribute is added to several elements and types defined in the XSD with a fixed value that is the SPM value. However, XML Schema syntax does not allow the addition of this attribute for element sequences defined in an xs:group construct nor is this attribute defined for string types that have to remain simple types, such as short and long identifiers. This attribute has no effect on XML validation, but it may be useful for applications, because an XML Schema processor has to make the fixed attribute and its value available to applications even if they are not explicitly specified in the COCD XML instance. This Standard does not define any conformance requirements regarding the use or interpretation of the spmattribute.

實作可能使用固定的spm屬性如下：

(1)舉例說明一份 XML 文件包含一個 spm 屬性的元件，那些 spm 屬性定義在一個 XML 架構處理器環境的 XSD 中。

(2)呼叫平常 XML 架構處理器的方法可以獲得 spm 屬性及其值。例如:使用 XPath 表達法，如〝./@ spm〞，將爲現行的元件存取屬性 spm。此屬性值可以在之後被用來比較帶有 SPM 的節點中值的長度。

An implementation might use the fixed spm attribute as follows:

—Instantiate an XML document that contains an element for which the spm attribute is defined in the XSD in an XML Schema processor environment.

—Get the spm attribute and its value by calling usual XML Schema processor methods. For example, using an XPath expression, such as "./@spm", will access the attribute spm for the current element. The attribute value can then be used to compare the length of the value in the node with the SPM.

D.2.4 XSD中機器可讀註解(Machine-readable annotations in the XSD)

W3C XML 架構定義語言允許包含註解，在 XSD 中預期是機器可讀的，藉由將此類的註解放入 xs:annotationelement 裡面的 xs：appInfo 元件中。XSD 包含的註解帶有關於對特定元件資料模型需求的資訊。

意識到國際規約的實作可能使用註解來找到連與元件關聯的 SPM，當 XSD 沒有提供 SPM 值作爲帶有一固定值之 spm 屬性時，或是實作可能會在 XSD 沒有提供蒐集型式屬性(此蒐集型式屬性帶有指出蒐集型式的固定值)時使用蒐集型式。本標準沒有定義任何相符的需求是關於註解或存在，使用，或解釋等表達在 XSD appInfo 元件中的資料。(爲了在本附錄中使範例讀起來更容易，註解已經移除多數的 XSD 片段。)

D.2.4 Machine-readable annotations in the XSD

The W3C XML Schema definition language allows the inclusion of annotations that are intended to be machine readable in an XSD by encapsulating such annotation in an xs:appInfo element inside an xs:annotationelement. The XSD contains annotations with information about requirements of the data model for particular elements.

An implementation that is aware of this notational convention may use an annotation to discover the SPM associated with an element when the XSD does not provide an SPM value as an spm attribute with a fixed value, or the implementation may be able to use the collection type when the XSD does not provide a collectionType attribute with a fixed value indicating the type of collection. This Standard does not define any conformance requirements regarding annotations or the existence, use, or interpretation of the data expressed in appInfo elements in the XSD. (To make the examples in this annex more readable, the annotations have been removed from most XSD fragments.)

在appInfo屬性元件中用到的元件名稱如下：

(1) spm：此名稱代表對元件的SPM值

(2)蒐集型式(collectionType)：此名稱代表用xs：sequence編碼的蒐集型式。相關的值有紀錄袋，陣列或集合。

理想上，這些元件將在一些具體說明的額外名稱空間中被定義，但是建立此名稱空間不在本標準的範圍內。同時名稱空間必須提供一份個別文件，又XSD在此文件外將無法被使用。參照外部、任意的名稱空間將複雜化XSD的部署及使用。爲了避免這樣的複雜性，但是依舊允許機器可讀的呈現在W3C XML架構推薦的精神中，資料模型的資訊資料是〝隱藏的〞在appInfo元件的評論中。爲了發現SPM或XSD定義中對元件的蒐集型式，實作能夠

(1)將從包含在appInfo元件中字串而來的XML片段實例化。

(2)將從包含在評論中字串而來的XML片段實例化。

(3)獲得spm 或收集型式(spmor collectionType)元件，一旦呈現就檢查它的值。

The element names used within appInfo annotation elements are as follows:

— spm: This name denotes the SPM value for the element.

— collectionType: This name denotes the type of collection encoded using xs:sequence. The associated value is one of bag, array, or set.

Ideally, such elements would be defined in some additional namespace specific to the data model, but creating such a namespace is outside of the scope of this Standard. Also, the name-space would have to be provided as a separate document, and the XSD would not be usable without the document. Referencing an external, arbitrary namespace would complicate deploying and using the XSD. To avoid such complications but still allow a machine-readable representation in the spirit of the W3C XML Schema recommendations, the data-model information data are “hidden” within a comment within an appInfo element. To discover the SPM or collection type for an element defined in the XSD, an implementation can

— Instantiate an XML fragment from the string contained in the appInfo element, if present, and extract

the comment from that fragment.

— Instantiate an XML fragment from the string contained in the comment.

— Get the element spm or collectionType, if present, and inspect its value.

D.3 6.1教材物件溝通編碼(Encoding of 6.1 Content object communication)

元件於資料模型中被載明爲 content_object_communication : record ，在XSD中利用cocdType型式之code元件表示。

符合 COCD XML 格式之實例常會被用來實作成 cocd 元件，此元件會紀錄著 content object communication record 並存放於 XML 之根元件中。

每個存在於複雜型式之 cocdType 元件最多只能出現一次，且此元件都是非必須且可選擇的項目。雖然此元件在 IEEE 1484.11.1-2004 標準之 6.1 章節中是依在 XSD 的次序作排列，但此元件還是可以以不同的順序出現。

```
<xs:complexType name="cocdType">
    <xs:all>
        <xs:element ref="commentsFromLearner" minOccurs="0"/>
            <xs:element ref="commentsFromLMS" minOccurs="0"/>
            ...
          <xs:element ref="timeLimitAction" minOccurs="0"/>
        <xs:element ref="totalTime" minOccurs="0"/>
    </xs:all>
</xs:complexType>
```

D.3 Encoding of 6.1 Content object communication

The element specified in the data model as content_object_communication : record is represented in the XSD by the element cocd of type cocdType.

A conforming COCD XML instance is expected to implement a cocd element as the root element of the XML document or fragment that contains a content object communication record.

Every element in the complex type cocdType is optional and may appear at most one time in a COCD XML instance. Although the elements are listed in the XSD in the same order as in 6.1 of IEEE Std 1484.11.1-2004, the elements may appear in any order.

每一個在 cocdType 型式中之元件都是參照頂層 XSD 所定義而來的。這些最上層的元件會在 XSD 中以相同的次序定義出現。

Every element in the cocdType type definition references an element defined at the top level of the XSD. These top-level elements appear in the XSD in the same order as in this type definition.

D.3.1 6.1.1學習者註釋編碼(Encoding of 6.1.1 Comments from learner)

此資料模型於XSD中被實作成commentsFromLearner元件，此元件可以出現次數爲“0”次 或是 “多

次”。commentFromLearner元件都爲commentType型式，並且被定義於IEEE 1484.11.1-2004標準之6.2.1章節中。

```
<xs:element name="commentsFromLearner">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="commentFromLearner" type="commentType"
       minOccurs="0" maxOccurs="unbounded">
  </xs:complexType>
</xs:element>
```

D.3.1 Encoding of 6.1.1 Comments from learner

This data-model element is implemented in the XSD as the element commentsFromLearner, which is a sequence of zero or more commentFromLearner elements. Each commentFromLearner element is of type commentType, which is an encoding of the comment type defined in 6.2.1 of IEEE Std 1484.11.1-2004.

D.3.2 6.1.2學習管理平台註釋編碼(Encoding of 6.1.2 Comments from LMS)

此資料模型於XSD中被實作成commentsFromLMS元件，此元件可以出現次數爲“0”次 或是 “多次”。commentFromLMS元件都是commentType型式，並且被定義於IEEE 1484.11.1-2004標準之6.2.1章節中。

```
<xs:element name="commentsFromLMS">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="commentFromLMS" type="commentType"
       minOccurs="0" maxOccurs="unbounded">
    </xs:complexType>
</xs:element>
```

D.3.2 Encoding of 6.1.2 Comments from LMS

This data-model element is implemented in the XSD as the element commentsFromLMS, which is a sequence of zero or more commentFromLMS elements. Each commentFromLMS element is of type commentType, which is an implementation of the comment type defined in 6.2.1 of IEEE Std 1484.11.1-2004.

D.3.3 6.1.3完成狀態編碼(Encoding of 6.1.3 Completion status)

此資料模型於XSD中被實作成Completion元件，其型式爲completionStatusType，此元件定義於IEEE 1484.11.1-2004標準之6.2.2章節中。

```
<xs:element name="completionStatus" type="completionStatusType"/>
```

D.3.3 Encoding of 6.1.3 Completion status

This data-model element is implemented in the XSD as the element completionStatus of type completionStatusType, which is an implementation of the completion status type defined in 6.2.2 of IEEE Std 1484.11.1-2004.

D.3.4 6.1.4完成門檻編碼(Encoding of 6.1.4 Completion threshold)

此資料模型於XSD中被實作成completionThreshold元件，其型式爲progressMeasureType，此元件定義於IEEE 1484.11.1-2004標準之6.2.7章節中。

```
<xs:element name="completionThreshold" type="progressMeasureType"/>
```

D.3.4 Encoding of 6.1.4 Completion threshold

This data-model element is implemented in the XSD as the element completionThreshold of type progressMeasureType, which is an implementation of the progress measure type defined in 6.2.7 of IEEE Std 1484.11.1-2004.

D.3.5 6.1.5學分編碼(Encoding of 6.1.5 Credit)

此資料模型於 XSD 中被實作成 credit 元件，其型式爲帶有標記值之列舉型式(enumerated type)，此元件定義於 IEEE 1484.11.1-2004 標準之 6.1.5 章節中。

```
<xs:element name="credit">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="credit"/>
      <xs:enumeration value="no_credit"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

D.3.5 Encoding of 6.1.5 Credit

This data-model element is implemented in the XSD as the element credit, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.5 of IEEE Std 1484.11.1-2004.

D.3.6 6.1.6資料模型版本編碼(Encoding of 6.1.6 Data model version)

此資料模型於XSD中被實作成dataModelVersion元件，其型式爲literalString250Type且爲一個自訂型式，主要的功能在於當其值爲空白時之避免突發性的修改狀況發生。此型式爲一個值爲250spm之屬性。

```
<xs:element name="dataModelVersion" type="literalString250Type"/>
```

D.3.6 Encoding of 6.1.6 Data model version

This data-model element is implemented in the XSD as the element dataModelVersion of type

literalString250Type, which is a custom type defined in the XSD to avoid accidental modification of the string value if it contains white space. This type has the fixed attribute spm with a value of 250.

D.3.7 6.1.7進入編碼(Encoding of 6.1.7 Entry)

此資料模型於XSD中被實作成entry元件，其型式爲帶有標記值之列舉型式(enumerated type)，此元件定義於IEEE 1484.11.1-2004標準之6.1.7章節中([1])。

```
<xs:element name="entry">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="ab_initio"/>
        <xs:enumeration value="resume"/>
        <xs:enumeration value=""/>
      </xs:restriction>
    </xs:simpleType>
</xs:element>
```

註([1]) 空字串（“”）呈現在COCD XML之實例中可以被視爲當作一個空元件(例如，<cocd><entry/ ></cocd>)。

D.3.7 Encoding of 6.1.7 Entry

This data-model element is implemented in the XSD as the element entry, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.7 of IEEE Std 1484.11.1-2004.

NOTE—An empty string ("") is represented in a COCD XML instance as an empty element (e.g., <cocd><entry/ ></cocd>).

D.3.8 6.1.8 退出編碼(Encoding of 6.1.8 Exit)

此資料模型於XSD中被實作成Exit元件，其型式爲帶有標記值之列舉型式(enumerated type)，此元件定義於IEEE 1484.11.1-2004標準之6.1.8章節中([1])。

```
<xs:element name="exit">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="logout"/>
        <xs:enumeration value="normal"/>
        <xs:enumeration value="suspend"/>
        <xs:enumeration value="timeout"/>
        <xs:enumeration value=""/>
      </xs:restriction>
    </xs:simpleType>
```

</xs:element>

註([1]) 一個空字串(“”)呈現在COCD XML之實例中可以被視爲當作一個空元件(例如，<cocd><exit/ ></cocd>)。

D.3.8 Encoding of 6.1.8 Exit

This data-model element is implemented in the XSD as the element exit, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.8 of IEEE Std 1484.11.1-2004.

NOTE—An empty string ("") is represented in a COCD XML instance as an empty element (e.g., <cocd><exit/></ cocd>).

D.3.9 6.1.9互動編碼(Encoding of 6.1.9 Interactions)

此資料模型於 XSD 中被實作成 Interactions 元件，其型式爲 interactionsType 之元件序列。此 interactionsType 型式被定義爲蒐集每一筆互動記錄之互動性元件。雖然此資料模型具體來說目的爲蒐集 interaction_type 之紀錄，但 XSD 於使用時必須以 xs：sequence 方式來使用，且允許可出現的次數不只一次。因此在此狀況中，xs：sequence 不應該被解釋成隱含任何特定次序。

```
<xs:complexType name="interactionsType">
  <xs:sequence>
    <xs:element name="interaction" type="interactionType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

D.3.9 Encoding of 6.1.9 Interactions

This data-model element is implemented in the XSD as a sequence of elements of type interactionType. The type interactionsType is defined as a collection of interaction elements, each of which represents an interaction record. Although the data model specifies that the collection is a bag of interaction_type records, the XSD has to use xs:sequence to allow multiplicity of the interaction element. In this case, xs:sequence should not be interpreted as implying any particular order.

D.3.9.1互動型式之實作(Implementation of interaction_type)

此資料模型結構於 XSD 中被實作成公用型式 InteractionType，包括了 Required elements identifier、type、optional elements objectiveIds、timeStamp、correctResponses、 weighting、learnerResponse、result、latency 以及 description 元件等。在呈現上，這些元件可以不用依照特定的次續出現。correctResponses 元件和 learnerResponse 元件被定義爲可重複使用之公用元件。

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

```
<xs:all>
  <xs:element name="identifier" type="longIdentifierType">
  <xs:element name="type" type="interactionTypeType"/>
  <xs:element name="objectiveIds" type="objectiveIdsType"
    minOccurs="0">
    <xs:unique name="uniqueInObjectivesIds">
      <xs:selector xpath="."/>
      <xs:field xpath="./objectiveId"/>
    </xs:unique>
  </xs:element>
  <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
  <xs:element name="correctResponses" type="correctResponsesType"
    minOccurs="0"/>
  <xs:element name="weighting" type="real7Type" minOccurs="0"/>
  <xs:element name="learnerResponse type"="learnerResponseType"
      minOccurs="0"/>
  <xs:element name="result" type="interactionResultType"
      minOccurs="0"/>
  <xs:element name="latency" type="timeIntervalType"
      minOccurs="0"/>
  <xs:element name="description" type="localizedString250Type"
      minOccurs="0"/>
</xs:all>
</xs:complexType>
```

D.3.9.1 Implementation of interaction_type

This data-model structure is implemented in the XSD by the global type interactionType, which includes the required elements identifier and type and the optional elements objectiveIds, timeStamp, correctResponses, weighting, learnerResponse, result, latency, and description. If present, these elements may occur in any order. The elements correctResponses and learnerResponse are defined as global elements to facilitate possible reuse.

D.3.9.2 6.1.9.1 識別符編碼(Encoding of 6.1.9.1 ID)

此資料模型於XSD中被實作成identifier元件，其型式爲longIdentifierType。

```
<xs:element name="identifier" type="longIdentifierType"/>
```

D.3.9.2 Encoding of 6.1.9.1 ID

This data-model element is implemented in the XSD as the element identifier of type longIdentifierType.

D.3.9.3 6.1.9.2型式編碼(Encoding of 6.1.9.2 Type)

此資料模型於XSD中被實作成Type元件，其型式爲interactionType。

```
<xs:element name="type" type="interactionTypeType"/>
```

D.3.9.3 Encoding of 6.1.9.2 Type

This data-model element is implemented in the XSD as the element type of type interactionTypeType.

D.3.9.4 互動型式(Interaction type type)

此資料模型結構於XSD中被實作成公用型式InteractionType，其主要被定義爲帶有標記值之列舉型式(enumerated type)，並定義於IEEE 1484.11.1-2004標準之6.1.9.2章節中。

```
<xs:simpleType name="interactionTypeType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="true_false"/>
    <xs:enumeration value="multiple_choice"/>
    <xs:enumeration value="fill_in"/>
    <xs:enumeration value="long_fill_in"/>
    <xs:enumeration value="likert"/>
    <xs:enumeration value="matching"/>
    <xs:enumeration value="performance"/>
    <xs:enumeration value="sequencing"/>
    <xs:enumeration value="numeric"/>
    <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>
```

D.3.9.4 Interaction type type

This data-model structure is implemented in the XSD as the global type interactionTypeType, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.9.2 of IEEE Std 1484.11.1-2004.

D.3.9.5 6.1.9.3目標識別符編碼(Encoding of 6.1.9.3 Objectives ID)

此資料模型於XSD中被實作成ObjectiveIds元件，其型式爲objectiveIdsType。此模型包含“0”個至“多個” longIdenitifierType元件，其有著獨特的應用限制。

```
<xs:element name="objectiveIds" type="objectiveIdsType"
  minOccurs="0">
   <xs:unique name="uniqueInObjectivesIds">
  <xs:selector xpath="./objectiveId"/>
  <xs:field xpath="."/>
  </xs:unique>
</xs:element>
```

D.3.9.5 Encoding of 6.1.9.3 Objectives ID

This data-model element is implemented in XSD as the element ObjectiveIds of type objectiveIdsType, which includes zero or more instances of the element longIdenitifierType. A uniqueness constraint is applied.

objectiveIdsType公用型式被定義爲蒐集longIdenitifierType型式之ObjectiveId元件。

```
<xs:complexType name="objectiveIdsType">
  <xs:sequence>
    <xs:element name="objectiveId" type="longIdentifierType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The global type objectiveIdsType is defined as a collection of ObjectiveId elements of type longIdenitifierType.

D.3.9.6 6.1.9.4時間點編碼(Encoding of 6.1.9.4 Time stamp)

此資料模型於 XSD 中被實作成 timestamp 元件，其型式爲 dateTimeTyp。

```
<xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
```

D.3.9.6 Encoding of 6.1.9.4 Time stamp

This data-model element is implemented in the XSD as the element timestamp of type dateTimeType.

D.3.9.7 6.1.9.5正確回應編碼(Encoding of 6.1.9.5 Correct responses)

此資料模型於 XSD 中被實作成 correctResponse 元件，其型式爲 correctResponsesTyp。

```
<xs:element name="correctResponses" type="correctResponsesType"
  minOccurs="0"/>
```

D.3.9.7 Encoding of 6.1.9.5 Correct responses

This data-model element is implemented in the XSD as the element correctResponses of type correctResponsesType.

D.3.9.8 正確回覆型式(Correct responses type)

此公用型式實作了不同的 correct_responses 之資料結構，並定義了不同型式的互動方式。無論是較具複雜性之互動資料結構，其實作方式即是選擇一個公用群組以實作於 SCD 中。COCD XML 實例的實作可選擇一個正確的群組來結合此互動型式。並用 W3C XML 之架構以及所定義語言來驗證。參考 COCD XML 實例之樣本

(見附錄 C)做爲呈現此型式之真實應用案例。

```
<xs:complexType name="correctResponsesType">
  <xs:choice>
  <xs:group ref="grpCorrectTrueFalse"/>
  <xs:group ref="grpCorrectMultipleChoice"/>
  <xs:group ref="grpCorrectFillIn"/>
  <xs:group ref="grpCorrectLongFillIn"/>
  <xs:group ref="grpCorrectLikert"/>
  <xs:group ref="grpCorrectMatching"/>
  <xs:group ref="grpCorrectPerformance"/>
  <xs:group ref="grpCorrectSequencing"/>
  <xs:group ref="grpCorrectNumeric"/>
  <xs:group ref="grpCorrectOther"/>
 </xs:choice>
</xs:complexType>
```

D.3.9.8 Correct responses type

This global type implements the various correct_responses data structures defined in the data model to correspond to different types of interactions. Regardless of the complexity of the data structure for a type of interaction, the choice is implemented in the XSD by a global group. It is up to the COCD XML instance implementation to choose the correct group to match the interaction type. This correspondence cannot be expressed or validated using the W3C XML Schema definition language. Refer to the sample COCD XML instance (see Annex C) for examples showing how this type can be used in an actual COCD XML instance.

這些群組在D.7中有詳細的描述。

These groups are described in detail in D.7.

D.3.9.9 6.1.9.6加權編碼(Encoding of 6.1.9.6 Weighting)

此資料模型於 XSD 中被實作成 weighting 元件，其型式爲 real7Type。

```
<xs:element name="weighting" type="real7Type" minOccurs="0"/>
```

D.3.9.9 Encoding of 6.1.9.6 Weighting

This data-model element is implemented in the XSD as the element weighting of type real7Type.

D.3.9.10 6.1.9.7學習者回覆的編碼(Encoding of 6.1.9.7 Learner response)

此資料模型於 XSD 中被實作成 learnerResponse 元件，其型式爲 learnerResponseType。

```
<xs:element name="learnerResponse" type="learnerResponseType"
```

minOccurs="0"/>

D.3.9.10 Encoding of 6.1.9.7 Learner response

This data-model element is implemented in the XSD as the element learnerResponse of type learnerResponseType.

D.3.9.11學習者回覆型式(Learner response type)

此公用型式，實作各種定義在資料模型中之學習者回覆(learner_response)資料結構，用來對應各種互動型式。無論互動型式資料結構如何複雜，選項都實作成XSD中的公用群組。由COCD XML實例的實作，決定選擇符合互動型式的正確群組。參照COCD XML實例(見附錄C)之範例以了解在真實的COCD XML實例中的典型使用方式。

```
<xs:complexType name="learnerResponseType">
  <xs:choice>
    <xs:group ref="grpResponseTrueFalse"/>
    <xs:group ref="grpResponseMultipleChoice"/>
    <xs:group ref="grpResponseFillIn"/>
    <xs:group ref="grpResponseLongFillIn"/>
    <xs:group ref="grpResponseLikert"/>
    <xs:group ref="grpResponseMatching"/>
    <xs:group ref="grpResponsePerformance"/>
    <xs:group ref="grpResponseSequencing"/>
    <xs:group ref="grpResponseNumeric"/>
    <xs:group ref="grpResponseOther"/>
  </xs:choice>
</xs:complexType>
```

D.3.9.11 Learner response type

This global type implements the various learner_response data structures defined in the data model to correspond to different types of interaction types. Regardless of the complexity of the data structure for a type of interaction, the choice is implemented in the XSD as a global group. It is up to the COCD XML instance implementation to choose the correct group to match the interaction type. Refer to the sample COCD XML instance (see Annex C) for examples of how this type can typically be used in an actual COCD XML instance.

這些群組在D.7中有詳細的描述。

These groups are described in detail in D.7.

D.3.9.12 6.1.9.8成果編碼(Encoding of 6.1.9.8 Result)

此資料模型於XSD中被實作成result元件，其型式爲interactionResultType。

```
<xs:element name="result" type="interactionResultType"
```

```
minOccurs="0"/>
```

D.3.9.12 Encoding of 6.1.9.8 Result

This data-model element is implemented in the XSD as the element result of type interactionResultType.

D.3.9.13 互動結果型式(Interaction result type)

此公用型式實作資料模型的必要條件，其結果值可以是一個對應於資料模型定義中允許之值或是一個數值的標記。此元件使用 xs：union 來取代 xs：choice 以避免必須在 XSD 中定義帶有固定名稱的次元件，而產生另一個巢狀結構。xs：union 的結構只允許單一值出現，以符合資料模型的必要條件。

```
<xs:simpleType name="interactionResultType">
   <xs:union memberTypes="real7Type interactionResultTokenType"/>
</xs:simpleType>
<xs:simpleType name="interactionResultTokenType">
  <xs:restriction base="xs:token">
     <xs:enumeration value="correct"/>
     <xs:enumeration value="incorrect"/>
     <xs:enumeration value="neutral"/>
     <xs:enumeration value="unanticipated"/>
  </xs:restriction>
</xs:simpleType>
```

D.3.9.13 Interaction result type

This global type implements the data-model requirement that the value of result may be a token that corresponds to a permissible value defined by the data model or a numeric value. This element uses xs:union instead of xs:choice to avoid having to define subelements with arbitrary names in the XSD, which would force another layer of nesting. The xs:union construct allows only a single value to occur, and thus it meets the requirement of the data model.

D.3.9.14 6.1.9.9潛伏時間編碼(Encoding of 6.1.9.9 Latency)

此資料模型於 XSD 中被實作成 latency 元件，其型式爲 timeIntervalType。

```
<xs:element name="latency" type="timeIntervalType" minOccurs="0"/>
```

D.3.9.14 Encoding of 6.1.9.9 Latency

This data-model element is implemented in the XSD as the element latency of type timeIntervalType.

D.3.9.15 6.1.9.10描述編碼(Encoding of 6.1.9.10 Description)

此資料模型於 XSD 中被實作成 description 元件，其型式爲 localizedString250Type。

```
<xs:element name="description" type="localizedString250Type"
```

minOccurs="0"/>

D.3.9.15 Encoding of 6.1.9.10 Description

This data-model element is implemented in the XSD as the element description of type localizedString250Type.

D.3.10啓動資料6.1.10之編碼(Encoding of 6.1.10 Launch data)

此資料模型於 XSD 中被實作成 launchData 元件，其型式爲 literalString4000Type，以避免包含空白時，字串值遭意外修改。

<xs:element name="launchData" type="literalString4000Type"/>

D.3.10 Encoding of 6.1.10 Launch data

This data-model element is implemented in the XSD as the element launchData of type literalString4000Type to avoid accidental modification of the string value if it contains white space.

D.3.11 6.1.11學習者ID編碼(Encoding of 6.1.11 Learner ID)

此資料模型於 XSD 中被實作成 learnerId 元件，其型式爲 longIdentifierType。

<xs:element name="learnerId" type="longIdentifierType"/>

D.3.11 Encoding of 6.1.11 Learner ID

This data-model element is implemented in the XSD as the element learnerId of type longIdentifierType.

D.3.12 6.1.12學習者名稱編碼(Encoding of 6.1.12 Learner name)

此資料模型於 XSD 中被實作成 learnerName 元件，其型式爲 localizedString250Type。

<xs:element name="learnerName" type="localizedString250Type"/>

D.3.12 Encoding of 6.1.12 Learner name

This data-model element is implemented in the XSD as the element learnerName of type localizedString250Type.

D.3.13 6.1.13學習者偏好資料編碼(Encoding of 6.1.13 Learner preference data)

此資料模型於 XSD 中被實作成 learnerPreferenceData 元件，其型式爲 learnerPreferenceType。

<xs:element name="learnerPreferenceData"
type="learnerPreferenceType" />

D.3.13 Encoding of 6.1.13 Learner preference data

This data-model element is implemented in the XSD as the element learnerPreferenceData of type

learnerPreferenceType.

此learnerPreferenceType公用型式實作IEEE 1484.11.1-2004標準6.1.13定義的learner_preference_type。它包含四個可能以任何次序(order)，音訊級別(audioLevel)，語言(language)，傳遞速度(deliverySpeed)，和音訊字幕(audioCaptioning)出現之可選擇的元件，以下會有更詳細地描述。

```
<xs:element name="learnerPreferenceData">
    <xs:complexType>
      <xs:all>
        <xs:element name="audioLevel" minOccurs="0">
          ...
        </xs:element>
        <xs:element name="language" type="languageType"
          minOccurs="0"/>
        <xs:element name="deliverySpeed" minOccurs="0">
          ...
        </xs:element>
        <xs:element name="audioCaptioning" minOccurs="0">
          ...
        </xs:element>
      </xs:all>
    </xs:complexType>
</xs:element>
```

The global type learnerPreferenceType implements the learner_preference_type defined in 6.1.13 of IEEE Std 1484.11.1-2004. It contains four optional elements that may appear in any order, audioLevel, language, deliverySpeed, and audioCaptioning, which are described in more detail below.

D.3.13.1 6.1.13.1音訊級別編碼(Encoding of 6.1.13.1 Audio level)

此資料模型於XSD中被實作成audioLevel元件，並定義成learnerPreferenceData內含的選擇性元件。它以real7Type型式爲基礎，加上大於或等於零的限制。

```
      <xs:element name="audioLevel" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="real7Type ">
            <xs:minInclusive value="0"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

D.3.13.1 Encoding of 6.1.13.1 Audio level

This data-model element is implemented in the XSD as the element audioLevel, which is defined inline as an optional element of learnerPreferenceData. It is based on the type real7Type, with a restriction that its value is greater than or equal to zero.

D.3.13.2 6.1.13.2語言編碼(Encoding of 6.1.13.2 Language)

此資料模型於 XSD 中被實作成 language 元件，此元件定義成 learnerPreferenceData 內含的選擇性元件。其型式爲 languageType。

```
<xs:element name="language" type="languageType" minOccurs="0"/>
```

D.3.13.2 Encoding of 6.1.13.2 Language

This data-model element is implemented in the XSD as the element language, which is defined inline as an optional element of learnerPreferenceData. It is of type languageType.

D.3.13.3 6.1.13.3傳遞速度編碼(Encoding of 6.1.13.3 Delivery speed)

此資料模型於 XSD 中被實作成 deliverySpeed 元件，此元件定義成 learnerPreferenceData 內含的選擇性元件。它以 real7Type 型式爲基礎，加上大於或等於零的限制。

```
<xs:element name="deliverySpeed" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="real7Type">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

D.3.13.3 Encoding of 6.1.13.3 Delivery speed

This data-model element is implemented in the XSD as the element deliverySpeed, which is defined inline as an optional element of learnerPreferenceData. It is based on the type real7Type, with a restriction that its value is greater than or equal to zero.

D.3.13.4 6.1.13.4音訊字幕編碼(Encoding of 6.1.13.4 Audio captioning)

此資料模型於 XSD 中被實作成 audioCaptioning 元件，此元件定義成 learnerPreferenceData 內的選擇性元件。其型式爲擁有一個標記值的列舉型式，該標記值對應於一個在 IEEE 1484.11.1-2004 標準 6.1.13.4 定義中被允許的值。

```
<xs:element name="audioCaptioning" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="off"/>
      <xs:enumeration value="no_change"/>
```

```
            <xs:enumeration value="on"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

D.3.13.4 Encoding of 6.1.13.4 Audio captioning

This data-model element is implemented in the XSD as the element audioCaptioning, which is defined inline as an optional element of learnerPreferenceData. It is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.13.4 of IEEE Std 1484.11.1-2004.

D.3.14 6.1.14課程狀態編碼(Encoding of 6.1.14 Lesson status)

此資料模型於XSD中被實作成lessonStatus元件，其型式爲legacyStatusType。

```
<xs:element name="lessonStatus" type="legacyStatusType"/>
```

D.3.14 Encoding of 6.1.14 Lesson status

This data-model element is implemented in the XSD as the element lessonStatus of type legacyStatusType.

D.3.15 6.1.15位置編碼(Encoding of 6.1.15 Location)

此資料模型於XSD中被實作成location元件，其型式爲literalString1000Typ，以避免在包含空白時，字串值遭意外修改。

```
<xs:element name="location" type="literalString1000Type"/>
```

D.3.15 Encoding of 6.1.15 Location

This data-model element is implemented in the XSD as the element location of type literalString1000Type to avoid accidental modification of the string value if it contains white space.

D.3.16 6.1.16最大允許時間編碼(Encoding of 6.1.16 Max time allowed)

此資料模型於XSD中被實作成maxTimeAllowed元件，其型式爲timeIntervalType。

```
<xs:element name="maxTimeAllowed" type="timeIntervalType"/>
```

D.3.16 Encoding of 6.1.16 Max time allowed

This data-model element is implemented in the XSD as the element maxTimeAllowed of type timeIntervalType.

D.3.17 6.1.17模式編碼(Encoding of 6.1.17 Mode)

此資料模型於 XSD 中被實作成 mode 元件，爲擁有一個標記值的列舉型式，該標記值對應於一個在 IEEE 1484.11.1-2004 標準 6.1.17 定義中被允許的值。

```
<xs:element name="mode">
```

```
<xs:simpleType>
  <xs:restriction base="xs:token">
    <xs:enumeration value="browse"/>
    <xs:enumeration value="normal"/>
    <xs:enumeration value="review"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

D.3.17 Encoding of 6.1.17 Mode

This data-model element is implemented in the XSD as the element mode, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.17 of IEEE Std 1484.11.1-2004.

D.3.18 6.1.18目標編碼(Encoding of 6.1.18 Objectives)

此資料模型於 XSD 中被實作成 objective 元件序列，其型式爲 objectiveType。雖然資料模型具體指明了這是一個 objective 集合，但是 XSD 必須使用 xs：sequence 來允許多個 objective 元件。在此情況中，xs：sequence 不可被解釋爲隱含任何特定次序。在此集合中，每個 objective 元件內的強制性識別符元件的值必須是唯一的。

```
<xs:element name="objectives" type="objectivesType">
  <xs:unique name="uniqueInSetOfObjectives">
    <xs:selector xpath="./objective"/>
    <xs:field xpath="identifier"/>
  </xs:unique>
</xs:element>
```

公用型式的objectivesType，實作了objectiveType的objective元件集合。

```
<xs:complexType name="objectivesType">
  <xs:sequence>
    <xs:element name="objective" type="objectiveType"
     minOccurs="0"    maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:attribute name="spm" fixed="100"/>
</xs:complexType>
```

D.3.18 Encoding of 6.1.18 Objectives

This data-model element is implemented in the XSD as a sequence of objective elements of type objectiveType. Although the data model specifies that this is a set of objective, the XSD has to use xs:sequence to allow multiplicity of the objectiveelement. In this case, xs:sequence should not be

interpreted as implying any particular order. The value of the mandatory identifier element within each objective element in this collection has to be unique.

The global type objectivesType implements a collection of objective elements of type objectiveType.

D.3.18.1目標型式(Objective type)

此資料模型結構於 XSD 中被實作成 objectiveType 公用型式，包含必要的元件識別符及及可選擇性的分數(score)、狀態(status)、 進展測量(progressMeasure)、 完成狀態(completionStatus)、 成功狀態(successStatus)以及 描述(description)元件。如果有此等元件存在，則此元件可以任何次序出現。

```
<xs:complexType name="objectiveType">
  <xs:all>
    <xs:element name="identifier" type="longIdentifierType"/>
    <xs:element name="score" type="scoreType" minOccurs="0"/>
    <xs:element name="status" type="legacyStatusType"
      minOccurs="0"/>
    <xs:element name="progressMeasure"
      type="progressMeasureType" minOccurs="0"/>
    <xs:element name="completionStatus"
      type="completionStatusType" minOccurs="0"/>
    <xs:element name="successStatus"
      type="successStatusType" minOccurs="0"/>
    <xs:element name="description"
      type="localizedString250Type" minOccurs="0">
    </xs:element>
  </xs:all>
</xs:complexType>
```

D.3.18.1 Objective type

This data-model structure is implemented in the XSD as the global type objectiveType, which includes the required element identifier and the optional elements score, status, progressMeasure, completionStatus, successStatus, and description. If present, these elements may occur in any order.

D.3.19 6.1.19進展測量編碼(Encoding of 6.1.19 Progress measure)

此資料模型於 XSD 中被實作成 progressMeasure 元件，其型式爲 progressMeasureType。

```
<xs:element name="progressMeasure" type="progressMeasureType"/>
```

D.3.19 Encoding of 6.1.19 Progress measure

This data-model element is implemented in the XSD as the element progressMeasure of type progressMeasureType.

D.3.20 6.1.20原始及格分數編碼(Encoding of 6.1.20 Raw passing score)

此資料模型於 XSD 中被實作成 rawPassingScore 元件，其型式爲 real7Type。

<xs:element name="rawPassingScore" type="real7Type"/>

D.3.20 Encoding of 6.1.20 Raw passing score

This data-model element is implemented in the XSD as the element rawPassingScore of type real7Type.

D.3.21 6.1.21標度之及格分數編碼(Encoding of 6.1.21 Scaled passing score)

此資料模型於 XSD 中被實作成 scaledPassingScore 元件，其型式爲 scaledScoreType。

<xs:element name="scaledPassingScore" type="scaledScoreType"/>

D.3.21 Encoding of 6.1.21 Scaled passing score

This data-model element is implemented in the XSD as the element scaledPassingScore of type scaledScoreType.

D.3.22 6.1.22分數編碼(Encoding of 6.1.22 Score)

此資料模型於 XSD 中被實作成 score 元件，其型式爲 scoreType，該型式是由數個元件組成的複雜型式。

<xs:element name="score" type="scoreType"/>

D3.22 Encoding of 6.1.22 Score

This data-model element is implemented in the XSD as the element score of type scoreType, which is a complex type composed of several elements.

D.3.23 6.1.23連線時間編碼(Encoding of 6.1.23 Session time)

此資料模型於 XSD 中被實作成 sessionTime 元件，其型式爲 timeIntervalType。

<xs:element name="sessionTime" type="timeIntervalType"/>

D.3.23 Encoding of 6.1.23 Session time

This data-model element is implemented in the XSD as the element sessionTime of type timeIntervalType.

D.3.24 6.1.24成功狀態編碼(Encoding of 6.1.24 Success status)

此資料模型於 XSD 中被實作成 successStatus 元件，其型式爲 successStatusType。

<xs:element name="successStatus" type="successStatusType"/>

D.3.24 Encoding of 6.1.24 Success status

This data-model element is implemented in the XSD as the element successStatus of type successStatusType.

D.3.25 6.1.25懸置資料編碼(Encoding of 6.1.25 Suspend data)

此資料模型於 XSD 中被實作成 suspendData 元件，其型式爲 literalString4000Type，爲了避免在包含空白時，字串值遭意外修改。

```
<xs:element name="suspendData" type="literalString4000Type"/>
```

D.3.25 Encoding of 6.1.25 Suspend data

This data-model element is implemented in the XSD as the element suspendData of type literalString4000Type to avoid accidental modification of the string value if it contains white space.

D.3.26 6.1.26時間限制動作編碼(Encoding of 6.1.26 Time limit action)

此資料模型於 XSD 中被實作成 timeLimitAction 元件，其型式爲擁有一個標記值的列舉型式，且該標記值對應於一個在 IEEE 1484.11.1-2004 標準 6.1.26 定義中被允許的值。

```
<xs:element name="timeLimitAction">
    <xs:simpleType>
        <xs:restriction base="xs:token">
            <xs:enumeration value="continue_message"/>
            <xs:enumeration value="continue_no_message"/>
            <xs:enumeration value="exit_message"/>
            <xs:enumeration value="exit_no_message"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

D.3.26 Encoding of 6.1.26 Time limit action

This data-model element is implemented in the XSD as the element timeLimitAction, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.26 of IEEE Std 1484.11.1-2004.

D.3.27 6.1.27總時間編碼(Encoding of 6.1.27 Total time)

此資料模型於 XSD 中被實作成 totalTime 元件，其型式爲 timeIntervalType。

```
<xs:element name="totalTime" type="timeIntervalType"/>
```

D3.27 Encoding of 6.1.27 Total time

This data-model element is implemented in the XSD as the element totalTime of type timeIntervalType.

D.4 6.2 輔助資料型式編碼(Encoding of 6.2 Auxiliary data types)

輔助資料型式編碼在 D4.1 到 D4.10 中會被討論。

D.4 Encoding of 6.2 Auxiliary data types

The encodings of the data-model auxiliary data types are discussed in D.4.1 through D.4.10.

D.4.1 6.2.1 評論型式實作(Implementation of 6.2.1 Comment type)

此資料模型型式於 XSD 中被實作成 commentType 公用型式，其包含所需求的元件評論和可選擇的 location 和 timeStamp 元件。此元件則可以任何次序出現。

```
<xs:complexType name="commentType">
  <xs:all>
    <xs:element name="comment" type="localizedString4000Type"/>
    <xs:element name="location" type="literalString1000Type"
      minOccurs="0"/>
    <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

D.4.1 Implementation of 6.2.1 Comment type

This data-model type is implemented in the XSD as the global type commentType, which includes the required element comment and the optional elements location and timeStamp. If present, these elements may occur in any order.

D.4.2 6.2.2 完成狀態型式實作(Implementation of 6.2.2 Completion status type)

此資料模型型式於XSD中被實作成completionStatusType公用型式，定義爲擁有一個標記值的列舉型式，且該標記值對應於一個IEEE 1484.11.1-2004標準6.2.2定義中被允許的值。

```
<xs:simpleType name="completionStatusType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="completed"/>
    <xs:enumeration value="incomplete"/>
    <xs:enumeration value="not_attempted"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>
```

D.4.2 Implementation of 6.2.2 Completion status type

This data-model type is implemented in the XSD as the global type completionStatusType, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.2.2 of IEEE Std 1484.11.1-2004.

D.4.3 6.2.2 資料時間型式實作(Implementation of 6.2.3 Date time type)

此資料模型型式於XSD中被實作成dateTimeType公用型式，是建立在XML架構的原始資料型式dateTime。 dateTime型式是符合IEEE 1484.11.1-2004.1標準定義在6.2.3中具體指明需求的一個實作。

```
<xs:simpleType name="dateTimeType">
    <xs:restriction base="xs:dateTime"/>
</xs:simpleType>
```

dateTime 格式用下列形態來定義：

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

括號中的數值爲可選擇的。

D.4.3 Implementation of 6.2.3 Date time type

This data-model type is implemented in the XSD as the global type dateTimeType, which is based on the XML Schema primitive data type dateTime. The dateTime type is a conforming implementation of the requirements specified defined in 6.2.3 of IEEE Std 1484.11.1-2004.

The format for dateTime is defined by the following pattern:

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

where anything enclosed in square brackets is optional.

D.4.4 6.2.4 語言型式實作(Implementation of 6.2.4 Language type)

此資料模型型式於 XSD 中被實作成 languageType 公用型式，是建立在 XML 架構的衍生資料型式 languag)。language 型式是 IEEE 1484.11.1-2004.1 標準中 6.2.4 定義除了在 SPM 規格外，具體指明需求的一個符合之實作，將呈現在 XSD 的註解中。

```
<xs:simpleType name="languageType">
   <xs:restriction base="xs:language"/>
</xs:simpleType>
```

D.4.4 Implementation of 6.2.4 Language type

This data-model type is implemented in the XSD as the global type languageType, which is based on the XML Schema derived data type language. The language type is a conforming implementation of the requirements specified defined in 6.2.4 of IEEE Std 1484.11.1-2004 except for the specification of an SPM, which is represented in an annotation in the XSD.

D.4.5 6.2.5定位字串型式實作(Implementation of 6.2.5 Localized string type)

此資料模型型式於XSD中被實作成localizedStringType公用型式，建立在literalStringType公用型式的

基礎上。此基本型式被用於避免當隱含空白時，字串值遭意外的修改。此localizedStringType型式用來當作其他2種型式： (localizedString250Type和 localizedString4000Type的抽象基本型式，此2種型式皆隱含對spm屬性的一個不同的固定值。

```
<xs:complexType name="localizedStringType" abstract="true">
  <xs:simpleContent>
    <xs:extension base="literalStringType">
      <xs:attribute name="lang" type="languageType"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

D.4.5 Implementation of 6.2.5 Localized string type

This data-model type is implemented in the XSD as the global type localizedStringType, which is based on the global type literalStringType. That base type is used to avoid accidental modification of the string value if it contains white space. The type localizedStringType is used as an abstract base type for two other types, localizedString250Type and localizedString4000Type, each of which has a different fixed value for an spmattribute.

D4.6 6.2.6 長辨識符型式實作(Implementation of 6.2.6 Long identifier type)

此資料模型型式於XSD中被實作成longIdentifierType公用型式，其建立於XML架構的原始資料型式anyURI基礎中，anyURI型式爲IEEE 1484.11.1-2004.1標準中6.2.6定義除了SPM規格外，具體指明需求的一個符合之實作，將呈現在XSD的註解中。因爲此型式用在當簡單型式需求時，沒有SPM屬性可以附加上去的地方。

```
<xs:simpleType name="longIdentifierType">
  <xs:restriction base="xs:anyURI">
  </xs:restriction>
</xs:simpleType>
```

D.4.6 Implementation of 6.2.6 Long identifier type

This data-model type is implemented in the XSD as the global type longIdentifierType, which is based on the XML Schema primitive data type anyURI. The anyURI type is a conforming implementation of the requirements defined in 6.2.6 of IEEE Std 1484.11.1-2004 except for the specification of an SPM, which is represented in an annotation in the XSD. Because this type is used where a simple type is required, no spmattribute can be attached to it.

D.4.7 6.2.7進展測量型式之實作(Implementation of 6.2.7 Progress measure type)

此資料模型型式於XSD中被實作成progressMeasureType公用型式，建立在real7Type公用型式基礎上，也就是建立在XML架構允許的十進位資料型式基礎上。

```
<xs:simpleType name="progressMeasureType">
```

```
  <xs:restriction base="real7Type">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

D.4.7 Implementation of 6.2.7 Progress measure type

This data-model type is implemented in the XSD as the global type progressMeasureType, which is based on the global type real7Type that, in turn, is based on the XML Schema primitive data type decimal.

D.4.8 6.2.8分數型式實作(Implementation of 6.2.8 Score type)

此資料模型型式於XSD中被實作成scoreType公用型式，之中包含了可選擇的元件： scaled、 max、 min和 raw。這些元件可以任何次序出現。

```
<xs:complexType name="scoreType">
  <xs:all>
    <xs:element name="scaled" type="scaledScoreType"/>
    <xs:element name="max" type="real7Type" minOccurs="0"/>
    <xs:element name="min" type="real7Type" minOccurs="0"/>
    <xs:element name="raw" type="real7Type" minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

D.4.8 Implementation of 6.2.8 Score type

This data-model type is implemented in the XSD as the global type scoreType, which includes the optional elements scaled, max, min, and raw. If present, these elements may occur in any order.

D.4.9 6.2.9短識別符型式實作(Implementation of 6.2.9 Short identifier type)

此資料模型型式於XSD中被實作成shortIdentifierType公用型式，建立於XML架構anyURI原始資料型式中，anyURI型式爲IEEE 1484.11.1-2004.1標準中6.2.9定義除了SPM規格外，具體指明需求的一個符合之實作，將呈現在XSD的註解中。因爲此型式將用在當簡單型式要求時，沒有SPM屬性可以附加上去的地方。

```
<xs:simpleType name="shortIdentifierType">
  <xs:restriction base="xs:anyURI">
</xs:simpleType>
```

D.4.9 Implementation of 6.2.9 Short identifier type

This data-model type is implemented in the XSD as the global type shortIdentifierType, which is based on the XML Schema primitive data type anyURI. The anyURI type is a conforming implementation of the requirements defined in 6.2.9 of IEEE Std 1484.11.1-2004 except for the specification of an SPM,

which is represented in an annotation in the XSD. Because this type is used where a simple type is required, no spmattribute can be attached to it.

D.4.10 6.2.10成功狀態型式實作(Implementation of 6.2.10 Success status type)

此資料模型型式於XSD中被實作成successStatusType公用型式，其爲定義爲擁有一個標記值的列舉型式，且該標記值對應於一個在IEEE 1484.11.1-2004標準6.2.10定義中被允許的值。

```
<xs:simpleType name="successStatusType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="failed"/>
    <xs:enumeration value="passed"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>
```

D.4.10 Implementation of 6.2.10 Success status type

This data-model type is implemented in the XSD as the global type successStatusType, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.2.10 of IEEE Std 1484.11.1-2004.

D.5 其他文件資料型式實作(Implementation of other documented data types)

其他文件型式實作將在D5.1到D5.2中做討論。

D.5 Implementation of other documented data types

The implementations of other documented data types are discussed in D.5.1 through D.5.2.

D.5.1 real(10,7)實作(Implementation of real(10,7))

此型式的解釋定義於 IEEE 1484.11.1-2004 的 B.1 標準中。此資料模型型式於 XSD 中被實作成 real7Type 公用型式，建立在 XML 架構的十進位原始資料型式基礎上。

```
<xs:simpleType name="real7Type">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

透過並非限制部分或全部的位元數，XSD 在所有實用興趣的案例中提供了所需的精確度。

D.5.1 Implementation of real(10,7)

An explanation of this type is provided in B.1 of IEEE Std 1484.11.1-2004. This data-model type is implemented in the XSD as the global type real7Type, which is based on the XML Schema primitive data type decimal.

By neither restricting the number of fraction digits nor the number of digits in total, the XSD provides the required precision in all cases of practical interest.

D.5.2 區間時間資料型式實作(Implementation of the time interval data type)

此型式的解釋定義於 IEEE 1484.11.1-2004 標準的 B.2 標準中。此資料模型型式於 XSD 中被實作成 timeIntervalType 公用型式，建立在 XML 架構原始資料型式持續時間的基礎上。此持續時間型式爲資料模型需求中的一個相符合的實作。

```
<xs:simpleType name="timeIntervalType">
  <xs:restriction base="xs:duration"/>
</xs:simpleType>
```

此型式在 XSD 定義中並未對於第 2 部分的持續表現，進一步限制十進位位元的數目。

D.5.2 Implementation of the time interval data type

An explanation of this type is provided in B.2 of IEEE Std 1484.11.1-2004. This data-model type is implemented in the XSD as the global type timeIntervalType, which is based on the XML Schema primitive data type duration. The duration type is a conforming implementation of the requirements specified in the data model.

This type definition in the XSD does not enforce a restriction on the number of decimal digits for the seconds part of the duration expression.

D.6 其他XSD定義中的公用型式(Other global types defined in the XSD)

在D6.1到D6.4中的型式，在不只被1個XSD的元件或其他XSD定義的型式中建立區段所採用。

D.6 Other global types defined in the XSD

The types in D.6.1 through D.6.4 are used by more than one element in the XSD or as building blocks in the definition of other XSD types.

D.6.1 文字字串型式(Literal string type)

此型式是爲字串而定義的，使得當中的空白不會由於一個 XML 實作而改變。literalStringType 型式用來當作其他三種型式的一個基礎抽象型式，分別爲 literalString250Type、 literalString1000Type 和 literalString4000Type，每一個皆有 1 個對一 spm 屬性不同的固定值。

```
<xs:simpleType name="literalStringType">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="preserve"/>
  </xs:restriction>
</xs:simpleType>
```

D.6.1 Literal string type

This type is defined for strings in which white space should not be modified by an XML implementation. The type literalStringType is used as an abstract base type for three other types, literalString250Type, literalString1000Type, and literalString4000Type, each of which has a different fixed value for an spm attribute.

D.6.2 課程狀態型式(Lesson status type)

此資料模型型式被定義爲obsoleteStatusType公用型式，其定義爲擁有一個標記值的列舉型式，且標記值對應於一個在IEEE 1484.11.1-2004標準6.2.14定義中被允許的值。

```
<xs:simpleType name="obsoleteStatusType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="browsed"/>
    <xs:enumeration value="completed"/>
    <xs:enumeration value="failed"/>
    <xs:enumeration value="incomplete"/>
    <xs:enumeration value="not_attempted"/>
    <xs:enumeration value="passed"/>
  </xs:restriction>
</xs:simpleType>
```

D.6.2 Lesson status type

This data-model type is defined as the global type obsoleteStatusType, which is defined as an enumerated type with token values that correspond to the permissible values defined in 6.1.14 of IEEE Std 1484.11.1-2004.

D.6.3 標度分數型式（Scaled score type）

此資料模型型式被定義爲建立在real7Type型式基礎上的數值型式，受到定義在 IEEE 1484.11.1-2004標準中6.1.21的範圍限制。

```
<xs:simpleType name="scaledScoreType">
  <xs:restriction base="real7Type">
    <xs:minInclusive value="-1"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

D.6.3 Scaled score type

This data-model type is defined as a numeric type based on real7Type with the range constraints defined in 6.1.21 of IEEE Std 1484.11.1-2004.

D.6.4 XML特定的文字字串型式（Literal string type—XML specific）

此型式並未明確地定義在 IEEE 1484.11.1-2004 標準中，卻被隱含在 ISO/ IEC 11404:1996 [B2]中的字元字串資料型式中。因爲 XML 架構處理器可能修改一個字串值中的空白，所以需要具體指明在 XSD 中這些 IEEE 1484.11.1-2004 標準 6.1 中所定義的各式的、以字串爲基礎的元件值不可以被改變。藉由 literalStringType 型式，它指定必須要保留空白。此型式被用來當作基礎的型式進一步定義 localizedStringType 型式，就如同下列三種型式： literalString250Type、literalString1000Type、和 literalString4000Type，此三種型式各自帶有 250、1000 和 4000 的 SPM 屬性值。

```
<xs:simpleType name="literalStringType">
   <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
   </xs:restriction>
</xs:simpleType>
```

D.6.4 Literal string type—XML specific

This type is not defined explicitly in IEEE Std 1484.11.1-2004, but it is implicit in references to the ISO/ IEC 11404:1996 [B2] characterstring data type. Because XML Schema processors may modify the white space in a string value, it is necessary to specify in the XSD that the values for various string-based elements defined in 6.1. of IEEE Std 1484.11.1-2004 cannot be modified. This is done by defining the type literalStringType that specifies that white space must be preserved. This type is used as the base type to define the type localizedStringType as well as the three types literalString250Type, literalString1000Type, and literalString4000Type with fixed spm attribute values of 250, 1000, and 4000, respectively.

D.7 元件和群組用來實作回覆資料（Elements and groups used to implement response data）

在 D7.1 到 D7.21 中討論的全域且可重複使用的 XML 元件和群組，是用在 XSD 中，作爲學習者正確回覆的複雜、資料模型型式之定義的一部份。他們按照英文字首排列。這些 XSD 的片段透過觀看 COCD XML 實例（見附錄 C）可以更加容易理解。

D.7 Elements and groups used to implement response data

The globally defined, reusable XML elements and groups discussed in D.7.1 through D.7.21 are used in the XSD as parts of the definitions for the complex, data-model types for correct responses and learner response. They are listed in alphabetic order. These XSD fragments are easier to understand by looking at the sample COCD XML instance (see Annex C).

D.7.1 填寫正確回覆（Correct responses for fill-in）

在此群組定義下，每一個fillMatches元件皆爲一個先前定義的配對字串的序列。xs:sequence的用法必須允許多於一個的fillMatche集合實例。且不應被解釋成隱含任何次序。然而，每個fillMatches元件的xs:sequence中，matchText元件的次序是有意義的。W3C XML架構定義語言並沒有規定需要在

xs:sequence解釋上表元出此差異。實作上應該要知曉資料模型的必要條件。

```
<xs:group name="grpCorrectFillIn">
  <xs:sequence>
    <xs:element name="fillMatches" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="matchText"
            type="localizedString250Type" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="caseMatters"
          type="trueFalseType" use="optional" default="false"/>
        <xs:attribute name="orderMatters" type="trueFalseType"
          use="optional" default="true"/>
        <xs:attribute name="collectionType" fixed="array"/>
        <xs:attribute name="spm" fixed="10"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:group>
```

D.7.1 Correct responses for fill-in

In this group definition, each fillMatches element is a sequence of predefined matching strings. The use of xs:sequenceis required to allow more than one instance of a set of fillMatches. It should not be interpreted as implying any particular order. However, the order of the matchText elements within the xs:sequence for each fillMatches element is significant. The W3C XML Schema definition language has no provision to express this difference of interpretation of xs:sequence. Implementations should be aware of the data-model requirements.

D.7.2 對likert正確回覆(Correct responses for likert）

在此群組的定義中，likert 的選擇一個單獨、選擇性的辨識符。

```
<xs:group name="grpCorrectLikert">
  <xs:sequence>
    <xs:element name="choice" type="shortIdentifierType"
      minOccurs="0"/>
    </xs:sequence>
  </xs:group>
```

D.7.2 Correct responses for likert

In this group definition, the likert choice is a single, optional identifier.

D.7.3 對long fill-in正確回覆（Correct responses for long fill-in）

在此群組定義中，每一個matchText元件皆爲一個單獨、已定義之配對字串，該字串擁有一個選擇性的屬性，用來表示在此配對中，字母的大小寫是否有關。xs:sequence的用法必須允許多於一個的matchText集合實例。且不應被解釋成隱含任何次序。

```
<xs:group name="grpCorrectLongFillIn">
  <xs:sequence>
    <xs:element name="matchText" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="localizedString4000Type">
            <xs:attribute name="caseMatters"
              type="trueFalseType" use="optional" default="false"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:group>
```

D.7.3 Correct responses for long fill-in

In this group definition, each matchText element is a single, predefined, matching string with an optional attribute specifying whether case matters for this match. The use of xs:sequence is required to allow more than one instance of a set of matchText. It should not be interpreted as implying any particular order.

D.7.4 對配對正確回覆(Correct responses for matching）

在此群組定義中，每一個回覆 MatchingType 型式的 matchPattern 元件，都代表一個配對組的集合。xs:sequence 的用法必須允許多於一個的 matchPattern 實例。且不應被解釋成隱含任何次序。

D.7.4 Correct responses for matching

In this group definition, each matchPattern element of type response MatchingType represents a set of matching pairs. The use of xs:sequence is required to allow more than one instance of matchPattern. It should not be interpreted as implying any particular order.

```
<xs:group name="grpCorrectMatching">
  <xs:sequence>
    <xs:element name="matchPattern" type="matchingPairsType"
      minOccurs="0" maxOccurs="unbounded"/>
```

```
    </xs:sequence>
</xs:group>
```

matchingPairsType公用型式實作一個配對形態，此配對形態爲配對組的集合。其中不隱含次序。配對組不必是唯一。

The global type matchingPairsType implements a match pattern, which is a collection of matching pairs. No order is implied. Pairs need not be unique.

```
<xs:complexType name="matchingPairsType">
  <xs:sequence>
    <xs:element name="pair" type="matchingPairType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

matchingPairType公用型式實作一個單獨的配對組，並定義成擁有來源和目標屬性的空元件。

The global type matchingPairType implements a single matching pair, which is defined as an empty element with source and target attributes.

```
  <xs:complexType name="matchingPairType">
    <xs:attribute name="source" type="shortIdentifierType"/>
    <xs:attribute name="target" type="shortIdentifierType"/>
  </xs:complexType>
```

D.7.5 對多重選擇正確回覆（Correct responses for multiple choice）

此元件在資料模型中定義爲短辨識符集合，每一個都代表一個選擇。外部集合爲一 choice 元件的集合。xs:sequence 必須允許多於一個選項集合的實例。且不應解釋爲隱含任何特定的次序。

D.7.5 Correct responses for multiple choice

This element is defined in the data model as sets of short identifiers, each of which represents a choice. The outer set is a collection of choices elements. The use of xs:sequence is required to allow more than one instance of a set of choices. It should not be interpreted as implying any particular order.

```
<xs:group name="grpCorrectMultipleChoice">
  <xs:sequence>
    <xs:element ref="choices" minOccurs="0"/>
  </xs:sequence>
```

```
</xs:group>
```

choices公用元件爲bagOfChoiceTypes型式，爲choice元件的集合。xs:unique必須限制集合中爲單獨的choice元件的集合。

The global element choices is of type bagOfChoiceTypes, which is a collection of choice elements. The use of xs:unique is required to constrain the collection to a set of unique choice elements.

```
<xs:element name="choices" type="bagOfChoicesType">
   <xs:unique name="uniqueInChoicesIds">
      <xs:selector xpath="./choice"/>
      <xs:field xpath="."/>
   </xs:unique>
</xs:element>
```

bagOfChoiceTypes 型式爲 choice 元件的集合，xs:squence 不應解釋爲隱含任何特定的次序。

The type bagOfChoiceTypes is a collection of choice elements. The use of xs:sequence should not be interpreted as implying any particular order.

```
<xs:complexType name="bagOfChoicesType">
   <xs:sequence>
      <xs:element name="choice" type="shortIdentifierType"
         minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

D.7.6 對數值正確回覆（Correct responses for numeric）

此群組定義包含兩個選擇性元件，用來指出正確回覆之最小和最大值。

```
<xs:group name="grpCorrectNumeric">
   <xs:sequence>
      <xs:element name="min" type="real7Type" minOccurs="0"/>
      <xs:element name="max" type="real7Type" minOccurs="0"/>
   </xs:sequence>
</xs:group>
```

D.7.6 Correct responses for numeric

This group definition contains two optional elements that specify the min and max values for the correct response.

D.7.7 對其他正確回覆（Correct response for other）

此群組的定義包含一個一般的、文字字串的元件。

```
<xs:group name="grpCorrectOther">
   <xs:sequence>
      <xs:element name="correctOther" type="literalString4000Type"/>
   </xs:sequence>
</xs:group>
```

D.7.7 Correct response for other

This group definition contains a generic, literal-string element.

D.7.8 對表現正確回覆（Correct response for performance）

在此群組定義中，每一個 correctPerformancePatternType 型式的 performancePattern 元件都代表了一個已定義且帶有資料值的步驟序列。xs:sequence 必須允許多於一個 performancePattern 的實例。且不應該被解釋成隱含任何特定的次序。

D.7.8 Correct response for performance

In this group definition, each performancePattern element of type correctPerformancePatternType represents a predefined sequence of steps with associated data. The use of xs:sequence is required to allow more than one instance of performancePattern. It should not be interpreted as implying any particular order.

```
<xs:group name="grpCorrectPerformance">
   <xs:sequence>
      <xs:element name="performancePattern"
         type="correctPerformancePatternType" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:group>
```

correctPerformancePatternType公用型式具體指明了單一次回覆形態的編碼，該回覆型式是一序列的步驟，擁有選擇性的、額外的、文字性的或數值性的answer元件，以標明每個部驟。每一步驟透過元件stepName的值來做辨識。此步驟的次序是有意義的。

The global type correctPerformancePatternType specifies the encoding of a single performance response pattern, which is a sequence of steps with an optional, additional, literal, or numeric answer element specified for each step. Each step is identified by the value of the element stepName. The order of steps is significant.

```
<xs:complexType name="correctPerformancePatternType">
  <xs:sequence>
    <xs:element name="step" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:all>
          <xs:element name="stepName" type="shortIdentifierType"
            minOccurs="0"/>
          <xs:element name="stepAnswer" minOccurs="0">
            <xs:complexType>
              <xs:choice>
                <xs:element name="literal"
                  type="literalString250Type" minOccurs="0"/>
                <xs:element name="numeric" minOccurs="0">
                  <xs:complexType>
                    <xs:attribute name="min" type="real7Type"/>
                    <xs:attribute name="max" type="real7Type"/>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="orderMatters" type="trueFalseType"
    use="optional" default="true"/>
  <xs:attribute name="collectionType" fixed="array"/>
  <xs:attribute name="spm" fixed="250"/>
</xs:complexType>
```

D.7.9 對序列正確回覆（Correct response for sequencing）

在此群組定義中，每一個 stepSequenceType 型式的 stepSequence 元件都代表了已定義之步驟序列。xs:sequence 必須允許多於一個 stepSequence 的實例。且不應被解釋爲隱含任何特定的次序。

D.7.9 Correct response for sequencing

In this group definition, each stepSequence element of type stepSequenceType represents a predefined sequence of steps. The use of xs:sequence is required to allow more than one instance of stepSequence. It should not be interpreted as implying any particular order.

```
<xs:group name="grpCorrectSequencing">
   <xs:sequence>
      <xs:element name="stepSequence" type="stepSequenceType"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>
```

stepSequenceType型式代表一個步驟序列。xs:sequence在此應解釋爲隱含特定的次序。

The type stepSequenceType represents a sequence of steps. The use of xs:sequence should be interpreted here as implying a specific order.

```
<xs:complexType name="stepSequenceType">
   <xs:sequence>
      <xs:element name="step" type="shortIdentifierType" minOccurs="0"
         maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="collectionType" fixed="array"/>
   <xs:attribute name="spm" fixed="36"/>
</xs:complexType>
```

D.7.10 對真假正確回覆（Correct response for true false）

在此群組定義中，真-假的選項爲一帶有真值或假值的標記值。

```
<xs:group name="grpCorrectTrueFalse">
   <xs:sequence>
      <xs:element name="trueOrFalse" type="trueFalseType"/>
   </xs:sequence>
```

In this group definition, the true-false choice is a token with the value true or the value false.

trueFalseType公用型式實作互動型式的真假選項，該互動型式定義於IEEE 1484.11.1-2004標準的6.1.9.5 和 6.1.9.7。此公用型式也用於標記其他多種布林元件的標記值。

```
<xs:simpleType name="trueFalseType">
   <xs:restriction base="xs:token">
      <xs:enumeration value="true"/>
      <xs:enumeration value="false"/>
   </xs:restriction>
</xs:simpleType>
```

The global type trueFalseType implements true and false options for the interaction type true_false

specified in 6.1.9.5 and 6.1.9.7 of IEEE Std 1484.11.1-2004. This global type is also used for tokens for various other Boolean elements.

D.7.11 填入學習者回覆（Learner response for fill-in）

在此群組定義中，學習者回覆由零或多個 localizedString250Type 型式的 fillString 元件的序列所組成。

```
<xs:group name="grpResponseFillIn">
   <xs:sequence>
      <xs:element name="fillString" type="localizedString250Type"
         minOccurs="0" maxOccurs="unbounded">
      </xs:element>
   </xs:sequence>
</xs:group>
```

D.7.11 Learner response for fill-in

In this group definition, the learner response consists of a sequence of zero or more fillString elements of type localizedString250Type.

D.7.12 學習者對likert回覆（Learner response for likert）

在此群組定義中，學習者回覆爲一單獨的、可選擇的辨識符。

```
<xs:group name="grpResponseLikert">
   <xs:sequence>
      <xs:element name="choice" type="shortIdentifierType"
         minOccurs="0"/>
   </xs:sequence>
</xs:group>
```

D.7.12 Learner response for likert

In this group definition, the learner response is a single, optional identifier.

D7.13 學習者對long fill-in回覆（Learner response for long fill-in）

在此群組的定義中，學習者回覆爲一單一的定位字串。

```
<xs:group name="grpResponseLongFillIn">
   <xs:sequence>
      <xs:element name="longFillString" type="localizedString4000Type"
         minOccurs="0"/>
    </xs:sequence>
</xs:group>
```

D.7.13 Learner response for long fill-in

In this group definition, the learner response is a single localized string.

D.7.14 學習者對配對回覆（Learner response for matching）

在此群組的定義中，學習者回覆爲一配對的形態，該配對形態是配對組的集合。不隱念任何次序。配對組不需要是唯一的。

```
<xs:group name="grpResponseMatching">
   <xs:sequence>
      <xs:element name="matchPattern" type="matchingPairsType"/>
   </xs:sequence>
</xs:group>
```

D.7.14 Learner response for matching

In this group definition, the learner response is a match pattern, which is a collection of matching pairs. No order is implied. Pairs need not be unique.

D.7.15 學習者對多重選擇回覆（Learner response for multiple choice）

在此群組定義中，學習者回覆爲一識別符串列，每一個識別符都具體指明了一個選擇。不應隱含任何次序。

```
<xs:group name="grpResponseMultipleChoice">
   <xs:sequence>
      <xs:element name="choices" type="setOfChoicesType"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseNumeric">
   <xs:sequence>
      <xs:element name="number" type="real7Type" minOccurs="0"/>
   </xs:sequence>
</xs:group>
<xs:group name="grpResponseOther">
   <xs:sequence>
      <xs:element name="responseOther" type="literalString4000Type"/>
   </xs:sequence>
</xs:group>
```

D.7.15 Learner response for multiple choice

In this group definition, the learner response is a list of identifiers, each of which specifies one choice. No order should be implied.

D.7.16 學習者對數值回覆（Learner response for numeric）

在此群組定義中，學習者回覆爲一單一的數值。

```
<xs:group name="grpResponseNumeric">
  <xs:sequence>
    <xs:element name="number" type="real7Type" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

D.7.16 Learner response for numeric

In this group definition, the learner response is a single numeric value.

D.7.17 學習者對其他回覆（Learner response for other）

在此群組定義中，學習者回覆爲一單一文字字串。

```
<xs:group name="grpResponseOther">
  <xs:sequence>
    <xs:element name="responseOther" type="literalString4000Type"
      minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

D.7.17 Learner response for other

In this group definition, the learner response is a single literal string.

D.7.18 學習者對表現回覆（Learner response for performance）

在此群組定義中，學習者回覆爲一單一次的表現形態，表示帶有相關資料的實際步驟序列。

D.7.18 Learner response for performance

In this group definition, the learner response is a single performance pattern that represents the actual sequence of steps with associated data.

```
<xs:group name="grpResponsePerformance">
  <xs:sequence>
    <xs:element name="step" type="learnerPerformanceStepType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>
```

learnerPerformanceStepType 型式實作一個表現步驟的學習者回覆之元件。

The type learnerPerformanceStepType implements the learner response elements for a single performance step.

```
<xs:complexType name="learnerPerformanceStepType">
  <xs:all>
    <xs:element name="stepName" type="shortIdentifierType"
      minOccurs="0"/>
    <xs:element name="stepAnswer" minOccurs="0">
      <xs:complexType>
        <xs:choice>
          <xs:element name="literal" type="literalString250Type"
            minOccurs="0"/>
          <xs:element name="numeric" type="real7Type"
            minOccurs="0"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:all>
</xs:complexType>
```

D.7.19 學習對序列回覆（Learning response for sequencing）

在此群組定義中，學習者回覆爲一識別符步驟之串列。

```
<xs:group name="grpResponseSequencing">
  <xs:sequence>
    <xs:element name="steps" type="stepSequenceType" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

D.7.19 Learning response for sequencing

In this group definition, the learner response is a list of steps identifiers.

D.7.20 學習者對真假回覆（Learner response for true false）

在此群組定義中，學習者回覆爲一個零或多個步驟識別符的序列。

```
<xs:group name="grpResponseTrueFalse">
  <xs:sequence>
    <xs:element name="trueOrFalse" type="trueFalseType"
      minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

D.7.20 Learner response for true false

In this group definition, the learner response is a sequence of zero or more step identifiers.

D.7.21配對回覆型式（Matching response type）

此型式實作一個配對組的集合。沒有特定次序隱含其中。

```
<xs:complexType name="responseMatchingType">
   <xs:sequence>
      <xs:element name="pair" type="matchingPairType"
         maxOccurs="unbounded">
   </xs:element>
</xs:sequence>
```

D.7.21 Matching response type

This type implements a set of matching pairs. No particular order is implied.

附錄 E

（資訊型）

網路可使用之 XSD 檔案與範例實例

在附錄B中的XSD檔案和在附錄C中的範例實例可在以下網址中取得：

http://standards.ieee.org/downloads/1484/1484.11.3-2005/ieee_1484.11.3-2005.xsd

此XSD檔案可以下載並直接包含於應用程式中。

Annex E

(informative)

Internet availability of the XSD file and example instance

The XSD file in Annex B and the example instance in Annex C are available on the World Wide Web at the following URL: http://standards.ieee.org/downloads/1484/1484.11.3-2005/ieee_1484.11.3-2005.xsd

The XSD file is available for downloading and for direct inclusion in applications.

英中名詞對照

- A -

array	陣列
attribute	屬性
audio captioning	音訊字幕

- B -

bag	紀錄袋
binding	繫結

- C -

cardinality	基數
character	字元
character string	字元串
content	教材
content object communication data (COCD)	教材物件溝通資料
convention	規約
credit	學分
collection	

- D -

Data model	資料模型

- E -

element	元件
enumerated type	列舉型式
Extensible Markup Language(XML)	可延伸標示語言

- F -

fragment	片段

- G -

- H -

- I -

identifier	識別符
instance	實例
interaction	互動

- J -

- K -

- L -

latency	潛時
literal	文字

- M -

max time allowed	最大允許時間

- N -

- O -

- P -

prefix	前綴

- Q–

- R -

raw passing score	原始及格分數
root element	根元件

- S -

scaled passing score	標度及格分數
schema	架構
session time	交談時間
SPM(smallest permitted maximum)	最小允許上限值
suspend data	懸置資料

- T -

time stamp	時戳

- U -

- V -

- W -

World Wide Web Consortium (W3C)	全球資訊網聯盟(W3C)

- X -

XML Schema definition (XSD)	XML 架構定義(XSD)

- Y -

- Z -

教材物件溝通資料模型之可延伸標示語言架構繫結 — 英文草案

CNS	Data Model for Content to Learning Management System Communication	General No..	xxx-x
		Classified No.. .	xxxx-x

Data Model for Content to Learning Management System Communication

Contents

1. Overview

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

1.1 Scope

This Standard describes a data model to support the interchange of agreed upon data elements and their values between a learning-related content object and a runtime service (RTS) used to support learning management. This Standard does not specify the means of communication between a content object and an RTS nor how any component of a learning environment shall behave in response to receiving data in the form specified. This Standard is based on a related data model defined in "Computer Managed Instruction (CMI) Guidelines For Interoperability," version 3.5 [B1], by the Aviation Industry CBT Committee (AICC).To balance the need to support existing implementations with the need to make technical corrections and support emerging practice, this Standard selectively includes those data elements from the CMI specification that are commonly implemented, renames some data elements taken from the CMI specification to clarify their intended meaning, modifies the data types of data elements taken from the CMI specification to reflect ISO standard data types and internationalization requirements, removes some organizational structures used in the CMI specification to group data elements that are specific to the AICC community of practice and not generally applicable, and introduces some data elements not present in the CMI specification to correct known technical deficiencies in data elements taken from that specification.

1.2 Purpose

There is widespread acknowledgement that the data model for content object communication defined in the AICC "Computer Managed Instruction (CMI) Guidelines for Interoperability," version 3.5 [B1], has broad applicability to systems used for learning management. The purpose of this Standard is to

build consensus around, resolve ambiguities in, and correct defects in the AICC data model for the data exchanged between learning-related content objects and an RTS used to support learning management.

2. Terms and definitions

2.1 Definitions

For purposes of this Standard, the following terms and definitions apply. The Authoritative Dictionary of IEEE Standards Terms [B2] should be referenced for terms not defined in this Clause.

2.1.1 content object: A collection of digital content that is intended for presentation to a learner by a learning technology system. It may include learning material and processing code. Example: A content object might be an HTML page with an embedded video clip and an ECMAScript component written in accordance with IEEE Std 1484.11.2TM-2003.

NOTE：For more information on IEEE Std 1484.11.2-2003, see [B3].

2.1.2 implementation defined (adj.): An indication that the implementation provider shall define and document the requirements for correct program constructs and correct data of a value or behavior. When the value or behavior in the implementation is designed to be variable or customizable on each instantiation of the system, the implementation provider shall document the nature and permissible ranges of this variation.

2.1.3 interaction: A recognized and recordable input or group of inputs from a learner to a content object.

2.1.4 launch (v.): To cause a content object to be delivered to a learner.

2.1.5 learner: An individual engaged with a learning technology system to acquire knowledge or skills.

2.1.6 learner attempt: A tracked effort by a learner to satisfy the requirements of a learning activity that uses a content object. It may span one or more learner sessions and be suspended between learner sessions.

See also: learner session.

NOTE：The learner attempt begins with the beginning of the first learner session and continues until the learning activity terminates.

2.1.7 learner session: An uninterrupted period of time during which a learner is accessing a content object.

See also: learner attempt.

2.1.8 learning management system (LMS): A computer system that may include the capabilities to register learners, schedule learning resources, control and guide the learning process, analyze and report learner performance, and schedule and track learners. See also: runtime service.

NOTE：Some implementations of learning management systems also have the

ability to launch and deliver content. For this Standard, these capabilities are known as a runtime service.

2.1.9 runtime service (RTS): Software that controls the execution and delivery of learning content and that may provide services such as resource allocation, scheduling, input–output control, and data management.

See also: learning management system.

2.1.10 score: A numerical value or a point on a descriptive scale. A score may be the result of a learner assessment.

2.2 Acroynms and abbreviations

AICC	Aviation Industry CBT Committee
CMI	computer managed instruction
IANA	Internet Assigned Numbers Authority
LMS	learning management system
RTS	runtime service
SPM	smallest permitted maximum
URI	Uniform Resource Identifier
URN	Uniform Resource Name

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) applies.

IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax.

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 8601:2000, Data elements and interchange formats—Information interchange—Representation of dates and times.

ISO/IEC 646:1991, Information technology—ISO 7-bit coded character set for information interchange.

ISO/IEC 10646-1, Information technology—Universal Multiple-Octet Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual Plane.

ISO/IEC 11404:1996, Information technology—Programming languages, their environments and system software interfaces—Language-independent datatypes.

4. Conformance

Conformance to this Standard is discussed in 4.1 – 4.6.

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 Data instances:

A conforming data instance shall be an instance of the data model as defined in 6.1.

4.2 Sending implementations:

A conforming sending implementation shall send data instances that conform to this Standard.

4.3 Receiving implementations:

A conforming receiving implementation shall accept data instances that conform to this Standard.

4.4 Repository implementations:

A conforming repository implementation shall accept, store, and provide data that conform to this Standard upon request.

4.5 Implementation-defined values:

The processing and meanings of values that are not specified by this Standard (e.g., sentinel, missing, and empty values) are implementation-defined.

NOTE：For example, bindings, application profiles, or implementations may specify the processing or meanings of default values or sentinel values for specific data elements. An application profile might specify that in the absence of another value, the default value for mode is normal.

4.6 Smallest permitted maximum values:

This Standard defines SPM values for data elements with data types that include bag, array, set, and characterstring. For these data elements, a receiving implementation or a repository implementation that conforms to this Standard shall accept and process at least that number of entries or characters specified by the SPM for the element and may accept and process a larger number.

NOTES

1. The intent is for the SPM values to cover most cases.
2. In this subclause, the meaning of “processing” is dependent on the nature of the application.
3. This Standard defines no provisions for how and whether a sending system can determine whether a receiving system can process more than the SPM for a particular data element.

5. Conceptual model (informative)

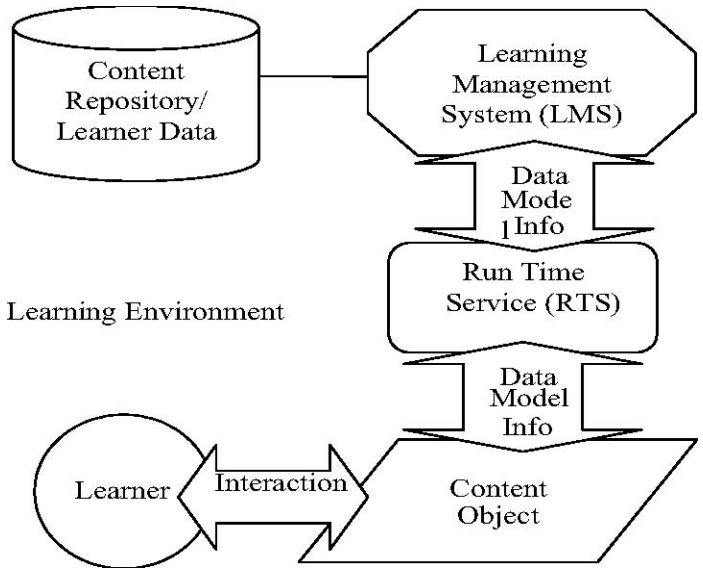
In one conceptual model for the use of this Standard, shown in Figure 1, the learner

interacts with a content object in the learning environment. The content object may require information about the learner. It acquires this information through an RTS, which, in turn, gets the information from an LMS.

As the learner interacts with the content object, the content object may gather information on the learner's performance. This information is passed to the RTS, which passes it on to the LMS.

Other conceptual models exist that may use the data model. Although this conceptual model includes an RTS and an LMS, they are not required for the use of the data model. This conceptual model is designed to describe only one possible use of the data model.

Figure 1—Conceptual model diagram



6. Data model

This Clause defines a data model that a content object can use for obtaining information from an RTS to enable the content object to perform its expected functions and that an RTS can use for obtaining information from a content object to enable the RTS to manage the content object properly. The data model provides a description of the information that may pass to and from the content object.

This Standard does not specify how, when, or in which direction the information may flow. In addition, this Standard does not specify persistence of the data, how often it may be written or rewritten, or by whom it may be created or destroyed.

Unless noted otherwise, all components of "records" are optional in a data instance.

NOTES

1. The use of ISO/IEC 11404 notation in the synopses in 6.1 and 6.2 is for descriptive purposes only. A complete implementation of the operations

defined in ISO/IEC 11404 is not required for conformance.

2. The ISO/IEC 11404 notation describes the semantics of the language-independent data types across all bindings(e.g., implementation of a data type as itself, its subtypes, its subclasses, and its specializations). For example, an ISO/IEC 11404 “array” may be implemented as an SQL table (because SQL tables support indexing, a requirement for arrays); an ISO/IEC 11404 “state” may be implemented as a C programming language bit field; an ISO/IEC 11404 “characterstring” may be implemented in any encoding (e.g., ISO 646, ASCII, ISO 8859-1, UTF-8, UTF-16, UTF-32, etc.) that supports the repertoire specified in the parameter for the characterstring data type.
3. All examples in 6.1 and 6.2 are informative and do not endorse any particular binding.
4. The following language-independent data types are defined in ISO/IEC 11404: array, bag, characterstring, choice, real, record, set, state, time, and timeinterval.
5. The data element and data type labels used in the synopses in 6.1 and 6.2 are for reference only. Implementations are not required to use the exact same labels, as long as the data elements and data types are semantically equivalent. It is recommended that implementations use spellings for labels similar to the spellings specified in this Standard.
6. This Standard does not define an extension mechanism for the data model. Implementers may create additional data models for content object communications. Such additional data models may be used to augment this data model to support different communities of practice.

6.1 Content object communication

(1) Synopsis

```
content_object_communication :
record
(

        comments_from_learner :
            array(0..249) of comment_type,
            // the SPM for the array is 250
            comments_from_lms :
            array(0..99) of comment_type,
            // the SPM for the array is 100
            completion_status :
            completion_status_type,
            completion_threshold :
```

real(10,7) range(0..1),
credit :
state(credit, no_credit),
data_model_version :
characterstring(iso-10646-1),
// SPM: 250 characters
entry :
state(ab_initio, resume, _nil_),
exit :
state(timeout, suspend, logout,
normal, _nil_),
interactions :
bag of interaction_type,
// SPM: 250 interaction_type
records in the bag
launch_data :
characterstring(iso-10646-1),
// SPM: 4000 characters
learner_id :
long_identifier_type,
learner_name :
localized_string_type(250),
// the parameter value is the
SPM
learner_preference_data :
learner_preference_type,
lesson_status :
state(passed, completed, failed,
incomplete, browsed,
not_attempted),

location :
characterstring(iso-10646-1),
// SPM: 1000 characters
max_time_allowed :
timeinterval(second,10,2),
mode :
state(browse, normal,
review),
objectives :

```
                set of objective_type,
                // SPM: 100 objective_type
                records in the bag
                progress_measure :
                progress_measure_type,
                raw_passing_score :
                real(10,7),
                scaled_passing_score :
                real(10,7) range(-1..1),
                score :
                score_type,
                session_time :
                timeinterval(second,10,2),
                success_status :
                success_status_type,
                suspend_data :
                characterstring(iso-10646-1),
                // SPM: 4000 characters
                time_limit_action :
                state( exit_message,
                continue_message,
                exit_no_message,
                      continue_no_message ),
                total_time :
                timeinterval(second,10,2),

),
```

(2) Description

The components of content_object_communication are defined in 6.1.1 – 6.1.27. Depending on the direction, destination, and purpose of the communication, an instance of content_object_communication shall include zero or more of the defined components.

6.1.1 Comments from learner

(1) Synopsis

```
            comments_from_learner :
                array(0..249) of comment_type,
                // the SPM for the array is 250
```

(2) Description

The values of this data element are comments from the learner.

Subclause 6.2.1 defines comment_type.

NOTES

1. The values of this data element are intended to provide feedback about the content object or the learning experience with the content object from a specific learner. Using this data element for other purposes may adversely affect interoperability.
2. This Standard does not specify the mechanism for collecting comments.

6.1.2 Comments from LMS

(1) Synopsis

```
comments_from_lms :
    array(0..99) of comment_type,
    // the SPM for the array is 100
```

(2) Description

The values of this data element are comments and annotations intended to be made available to the learner.

Subclause 6.2.1 defines comment_type.

NOTES

1. The values of this data element are intended to provide information about the content object or the learning experience with the content object. Using this data element for other purposes may adversely affect interoperability.
2. This Standard does not specify the mechanism for collecting comments.

6.1.3 Completion status

(1) Synopsis

```
completion_status :
    completion_status_type,
```

(2) Description

The value of this data element indicates whether the learner has completed the content object. Subclause 6.2.2 defines completion_status_type.

NOTE：This Standard does not specify how to determine completion_status. It may be reported by a content object, determined by an RTS by comparing a progress measure with a threshold, determined on the basis of objectives set by an outside agent (e.g., an instructor), or determined by some other means.

6.1.4 Completion threshold

(1) Synopsis

completion_threshold :

real(10,7) range(0..1),

(2) Description

The value of this data element is a value against which the measure of the progress the learner has made toward completing the content object can be compared to determine whether the content object should be considered completed.

NOTE：The completion_threshold is designed to be used in conjunction with the progress_measure (see 6.1.19). For example, if the completion_threshold for a content object is 0.85 and a learner achieves a progress_measure of 0.90, a completion_status of completed(see 6.1.3) may be assigned to that content object for that learner. However, this Standard does not specify or require that an RTS, content object, or any other system component, interpret or take action in response to a completion_threshold.

6.1.5 Credit

(1) Synopsis

credit :

state(credit, no_credit),

(2) Description

The value of this data element indicates whether the learner will be credited for performance in this content object. This data element shall have one of the following permissible values:

(a) credit: The learner is taking the content object for credit.

(b) no_credit: The learner is taking the content object for no credit.

NOTE：This Standard does not specify how to determine the value of credit or what it means to give credit for performance.

6.1.6 Data model version

(1) Synopsis

data_model_version :

characterstring(iso-10646-1),

// SPM: 250 characters

(2) Description

The value of this data element indicates the version of the data model defined in this Standard. The value shall consist of a period-delimited string containing major and minor release values as whole numbers, for example, “1.0”. Any characters appearing after the minor release value shall be separated from the minor release value by a period (“.”). The syntax and semantics of any characters following the minor release value are not specified by this Standard.

For this edition of this Standard, the major version number shall be “1” and the minor version number shall be “0”. Therefore, the first three characters of the value shall be “1.0”.
An implementation may append additional characters to the value of this data element, in which case, the first four characters shall be “1.0.”.

6.1.7 Entry

(1) Synopsis

entry :
state(ab_initio, resume, _nil_),

(2) Description

The value of this data element is information that asserts whether the learner has previously accessed the content object. This data element shall have one of the following permissible values:

(a) ab_initio: Indicates that the learner has not accessed the content object during the current learner attempt.
(b) resume: Indicates that (1) the learner has previously accessed the content object during the current learner attempt, and (2) upon exiting, the exit data element had the value suspend (see 6.1.8).
(c) _nil_: Indicates all other conditions. There is no knowledge of previous access, or no specific entry condition is indicated.

NOTE：If the value for entry is resume, it indicates that either location or suspend_data may contain data stored in a previous learner session that is relevant to resuming the learner session (see 6.1.15 and 6.1.25, respectively).

6.1.8 Exit

(1) Synopsis

exit :
state(timeout, suspend, logout, normal, _nil_),

(2) Description

The value of this data element indicates how or why the learner left the content object. This data element shall have one of the following permissible values:

(a) timeout: The content object terminated because the time limit specified by max_time_allowedhad been exceeded (see 6.1.16).
(b) suspend: The learner exited the content object with the intent of returning to it.
(c) logout: The content object signaled a desire to terminate the entire learning activity of which the content object is a part.
(d) normal: The content object exited normally.
(e) _nil_: The exit conditions are undetermined.

6.1.9 Interactions

(1) Synopsis

```
interactions :
    bag of interaction_type,
    // SPM: 250 interaction records in the bag

type interaction_type =
record
(

    id :
        long_identifier
        _type,
        type :

      state( true_false, multiple_choice, fill_in,
                    long_fill_in, likert, matching, performance,
                    sequencing, numeric, other ),

    objectives_id :
        array(0..9) of
        long_identifier_type,
        // the SPM for the array is 10
        time_stamp :
        date_time_type,
        correct_responses :
        correct_responses_type,
        weighting :
        real(10,7),
        learner_response :
        learner_response_type,

  result :
      choice
      (

          state( result_state, numeric ),
      )
      of
```

```
(

    result_state :
        state( correct, incorrect, unanticipated,
          neutral ),
        numeric :
        real(10,7),
        ),
        latency :
        timeinterval(second,10,2),
        description :
        localized_string_type(250),
        // the parameter value is the SPM

),
```

(2) Description

The values of this data element define information pertaining to an interaction for the purpose of measurement or assessment. An instance of an interaction_type record shall include an interaction identifier (see 6.1.9.1). If an instance includes either correct_responses or learner_response, then the instance shall include type (see 6.1.9.2). All other components are optional.

The components of interaction_type are defined in 6.1.9.1 – 6.1.9.10.

NOTES

1. Interactions are intended to be responses to individual questions or tasks that the content developer wants to record. This Standard does not specify how interaction data are to be recorded, used, or interpreted.
2. The interactions data model includes data elements that correspond to a limited set of interaction types, but it does not support logging of discrete learner events.
3. This Standard does not specify how interactions are presented or rendered.
4. This Standard does not specify how interactions are grouped in a question (i.e., one or multiple interactions per question).
5. The primary intent of interaction data is to communicate information about the status of an interaction object, such as a test item, a simulation, or another interactive feature of the content object. Interaction data also may be used to communicate interaction events as they occur, but in that case, only the data elements that carry information specific to the event should be communicated.

6.1.9.1 ID

(1) Synopsis

```
id :
    long_identifier_type,
```

(2) Description

The value of this data element is a label for the interaction. This label shall be unique at least within the scope of the content object.

Subclause 6.2.6 defines long_identifier_type.

NOTE：This Standard does not specify how IDs are created, assigned, or resolved.

6.1.9.2 Type

(1) Synopsis

type :

state(true_false, multiple_choice, fill_in, long_fill_in,

likert, matching, performance, sequencing, numeric,
other),

(2) Description

The value of this data element indicates which category of interaction is recorded in an instance of an interaction. It is also used to determine how the interaction response should be interpreted. This data element shall have one of the following permissible values. The content developer may create extended types using other.

(a) true_false: The interaction has two possible responses. Examples: “True or False,” “Yes or No,” and “Agree or Disagree.”

(b) multiple_choice: The interaction has a set of two or more possible responses. This interaction type can be used for interactions in which the learner selects just one choice and for interactions in which the learner can select more than one choice.

(c) fill_in: The interaction requires the learner to supply a short response in the form of one or more strings of characters. Note: Typically, the correct response consists of part of a word, one word, or a few words.

(d) long_fill_in: The interaction requires the learner to supply a response in the form of a long string of characters. Notes: (1) Typically, the correct response is a sentence, paragraph, or short composition, but long composition forms are also possible. (2) Typically, the interaction is presented as an examination statement the learner must analyze and respond to by creating a written answer of a specified length, such as a short or long essay.

(e) likert: The interaction asks the learner to select from a discrete set of choices on a scale. Note: This Standard does not specify the number of choices, the scale, or the meaning of the scale. Example: A typical response scale has five choices ranging from “strongly disagree” to “strongly agree.”

(f) matching: The interaction consists of two sets of items. Members of the first set are related to zero or more members of the second set. Responding to the interaction requires the learner to indicate matches between members of the first set and members of the second set.

(g) performance: The interaction requires the learner to perform a task that requires multiple steps. Example: The task is a simulation for the changing of a spark plug on an automobile engine involving six steps: (1) pull off the rubber boot from the plug, (2) unscrew the spark plug, (3) gap the replacement plug to a specific dimension, (4) screw in the replacement, (5) torque the plug using a torque wrench set to 12 foot-pounds, and (6) push the boot back on.

(h) sequencing: The interaction requires the learner to identify a logical order for the members of a list. Example: The learner may be asked to place a series of events in chronological order or to rank a group of items by the order of their importance.

(i) numeric: The interaction requires a numeric response from the learner.

(j) other: Any other type of interaction not defined by this Standard. The semantics and structure of the correct_responses and learner_response data element values are not defined by this Standard when the interaction type is other (see 6.1.9.5 and 6.1.9.7, respectively). Note: When the interaction type is other, information identifying this extended type should be embedded in the correct_responses and learner_response data element values. For example, this may take the form of a prefix in the string used to communicate those values.

6.1.9.3 Objectives ID

(1) Synopsis

```
objectives_id :
    array(0..9) of long_identifier_type,
        // the SPM for the array is 10
```

(2) Description

The values of this data element are labels for objectives (see 6.1.18) associated with the interaction. The labels shall be unique at least within the scope of the content object.

Subclause 6.2.6 defines long_identifier_type.

NOTE：This Standard does not specify how objective IDs are created, assigned, or resolved.

6.1.9.4 Time stamp

(1) Synopsis

```
time_stamp :
    date_time_type,
```

(2) Description

The value of this data element is the point in time at which the interaction was first made available to the learner for learner interaction and response.

Subclause 6.2.3 defines date_time_type.

NOTES

1.This Standard does not specify how the time_stamp value is obtained.

2.If several interactions are presented at the same time, they have the same time_stamp value. If an interaction was never available for response, such as an interaction that is not used in an adaptive test, no time_stamp value is available for that interaction.

3.If a time_stamp value is available for an interaction but no learner response data are available, it should be assumed that the interaction was made available to the learner but the learner did not respond.

6.1.9.5 Correct responses

(1) Synopsis

```
correct_responses :
correct_responses_type,

type correct_responses_type =
    choice
    (

        state( true_false, multiple_choice, fill_in, long_fill_in,
                likert, matching, performance, sequencing, numeric,
                other ),
    )
    of
    (

        true_false :
            state( true, false ),
            multiple_choice :

            set of set of short_identifier_type,
                // set of set SPM: 10 sets
                // set of short_identifier_type SPM: 36
                // short identifiers

            fill_in :
                bag of
                record
                // SPM: 5
```

```
                records
                (
                case_mat
                ters :

                    boolean,
                order_matters :
                boolean,

            match_text :

                array(0..9) of localized_string_type(250),
                    // the SPM for the array is 10
                    // the parameter value is the SPM for the
                    // localized string

                ),
            long_fill_in :
            bag of record
            // SPM: 5 records
            (
            case_matters :
            boolean,
            match_text :
            localized_string_type(
            4000),
            // the parameter value
            is the SPM
            ),

likert :
    short_identifier_type,
    matching :

  bag of bag of record
      // outer bag SPM: 5 inner bags
      // each inner bag SPM: 36 records

        (
            source :
```

```
short_identifier_type,
target :
short_identifier_type,
),
performance :
bag of record
// SPM: 5 records
(
order_matters :
boolean,
answers :
array(0..124) of record
// the SPM for the array is 125
(
step_name :
short_identifier_type,

   step_answer :
        choice
        (

             state( literal, numeric ),
        )
        of
        (

                  literal :
                       characterstring(iso-10646-1),
                       // SPM: 250 characters

             numeric :
                  record
                  (

           min :
                real(10,7),
                max :
                real(10,7),
                ),
                ),
```

```
                    ),
                    ),
                    sequencing :

    bag of array(0..35) of short_identifier_type,
        // bag SPM: 5 arrays
        // the SPM for the array is 36

numeric :
    record
    (

        min :
            real(10,7),
            max :
            real(10,7),
            ),

                other :
                    characterstring(iso-1046-1),
                    // SPM: 4000 characters

        ),
```

(2) Description

The values of this data element indicate the correct response(s) to the interaction. This data element shall have one of ten possible variants that shall match the conditions described below. The content developer may create extended types using other.

Several response types support more than one correct response. For these types, a list (bag) of correct response(s) is provided. A correct response may require multiple inputs. For these responses, a collection of input(s) is provided.

(a) true_false: A state that contains the values true and false. The state true means true or an equivalence of true in a particular context (e.g., agree, yes, richtig). The state false means false or an equivalence of false in a particular context (e.g., disagree, no, falsch).

(b) multiple_choice: A set that contains one or more sets of short identifiers. Any of the sets of short identifiers satisfies the requirement for a correct response. Multiple sets may be defined if more than one correct response exists. A set of short identifiers may contain zero or more short identifiers, all of which are required for a correct response. Each of the short identifiers represents an expected choice. If a set of short identifiers is empty, it represents that the correct response is no choice. Examples: (1) A single choice may be allowed: “alpha.” (2) Multiple sets of choices may be allowed: “alpha,” “bravo,” “charlie,”

and “alpha,” “bravo,” “delta.”

(c) fill_in: A bag of records. The bag contains one or more records, any of which satisfies the requirement for a correct response. Each record consists of an array of localized strings and two flags. The localized strings represent a correct response. The case_matters flag indicates whether the case of the string is used to evaluate the correctness of the response. If the value of the flag is true, the case of the learner response shall match the correct response. If the value of the flag is false, the case of the learner response is not used in evaluating the response. If case_matters is not specified, it is assumed to be false. The order_matters flag indicates whether the order of the inputs for a correct response matters. If the value of the flag is true, then order matters, and the order of the learner’s responses should be used to evaluate correctness of the response. If the value of the flag is false, then order does not matter, and the order of the learner’s responses should not be used to evaluate correctness of the response. If order_matters is not specified, it is assumed to be true.

(d) long_fill_in: A bag of records. The bag contains one or more records, any of which satisfies the requirement for a correct response. Each record consists of a localized string and a flag. The localized string represents a correct response. The flag indicates whether the case of the string is used to evaluate the correctness of the learner response. If the value of the flag is true, the case of the learner response shall match the correct response. If the value of the flag is false, the case of the learner response is not used in evaluating the response. If case_matters is not specified, it is assumed to be false. Note: Although a correct response for long_fill_in can be specified, the evaluation of a long_fill_in response typically involves an interpretative process that is outside of the scope of this Standard.

(e) likert: A short identifier that matches a choice on a scale. Note: Although a correct response for likert can be specified, likert interactions typically do not include correct responses.

(f) matching: A bag of bags of records. The single outer bag contains one or more inner bags. Each inner bag contains one or more records. If more than one inner bag exists, any of the inner bags satisfies the requirement for a correct response. If more than one record is contained by an inner bag, all records are required for the correct response specified by that inner bag. Each of the records is a pair of short identifiers representing an expected matching input. Each correct response pair consists of a source and a target. Each source and each target shall be represented by a short identifier. The scope for the short identifiers used for sources and targets shall be the interaction. The same short identifier may appear in more than one source-target pair.

(g) performance: A bag of records. The bag contains one or more records, any of which satisfies the requirement for a correct response. Each record consists of a flag and an array. The array represents a set of correct responses. The order_matters flag indicates whether the order of the inputs matters for a correct response. If the value of the flag is true, the order of the learner’s responses should be used to evaluate correctness of the response. If the value of the flag is false, the order of the learner’s responses should not be used to evaluate correctness of the response. If order_matters is not specified, it is assumed to be true. Each correct

response consists of a name and either a single literal value or a numeric range. If the correct response is expressed as a literal value, this Standard does not specify how to use the value to evaluate the corresponding response. If the correct response is expressed as a numeric range, the learner's response should be within the specified range to be judged correct.

(h) sequencing: A bag of arrays of short identifiers. The bag contains one or more arrays, any of which satisfies the requirement for a correct response. Each array represents a sequence of zero or more short identifiers for a correct response. Each short identifier identifies one element available to be sequenced when the interaction is presented to the learner. Each array shall contain a different sequence of short identifiers. Different arrays may contain different short identifiers.

(i) numeric: Two real numbers. The numbers may be used to express an inclusive range for the correct response. If a min value is specified with no max value, the upper limit of the range is unbounded. If a max value is specified with no min value, the lower limit of the range is unbounded. If both the min and max values are unspecified, both the upper and lower limits of the range are unbounded. If the min and max values are equal, the range is a single value. Note: This Standard does not specify the number of significant digits that should be considered in evaluating results against the specified range.

(j) other: A string defined by the specific "other" interaction type (see 6.1.9.2). The content of this string is not defined by this Standard.

Subclauses 6.2.5 and 6.2.9 define localized_string_type and short_identifier_type, respectively.

NOTE：The correct_responses data element is a structured mechanism for identifying the correct learner response or responses relating to each of the types of interactions described in 6.1.9.2. The determination of correctness is an implementation-defined feature (see 6.1.9.8).

6.1.9.6 Weighting

(1) Synopsis

weighting :
real(10,7),

(2) Description

The value of this data element is a weight given to the interaction that may be used by the content object to compute a value for a score.

NOTE：Interaction weights typically are used to explain the effect of an interaction on the value of the score data element for an objective or for the content object (see 6.1.18.2 and 6.1.22, respectively), but they are not intended to be used by systems other than the content object to compute a score.

6.1.9.7 Learner response

(1) Synopsis

```
learner_response :
    learner_response_type,

type learner_response_type =
    choice
    (

        state( true_false, multiple_choice, fill_in, long_fill_in,
                likert, matching, performance, sequencing, numeric,
                other ),

    )
    of
    (

        true_false :
            state( true, false ),
            multiple_choice :
            set of short_identifier_type,
            // SPM: 36 short identifiers
            fill_in :

         array(0..9) of localized_string_type(250),
             // the SPM for the array is 10
             // the parameter value is the SPM for the localized
             // string

           long_fill_in :
               localized_string_type(4000),
               // the parameter value is the
               SPM
               likert :
               short_identifier_type,
               matching :
               bag of record
               // SPM: 36 records
               (
               source :
               short_identifier_type,
               target :
```

```
short_identifier_type,
),
performance :
array(0..249) of record
// the SPM for the array is 250
(
step_name :
short_identifier_type,

step_answer :
    choice
    (

        state( literal, numeric ),
    )
    of
    (

literal :
    characterstring(iso-10646-1),
    // SPM: 250 characters
    numeric :
    real(10,7),
    ),
    ),
    sequencing :
    array(0..35) of
    short_identifier_type,
    // the SPM for the array is 36
    numeric :
    real(10,7),
    other :
    characterstring(iso-1046-1),
    // SPM: 4000 characters

),
```

(2) Description

The values of this data element consist of data generated when a learner responds to an interaction. This data element shall have one of the ten possible variants that shall match the conditions described

below. The content developer may create extended types using other.

(a) true_false: A state that contains the values true and false. The state true means true or an equivalence of true in a particular context (e.g., agree, yes, richtig). The state false means false or an equivalence of false in a particular context (e.g., disagree, no, falsch).

(b) multiple_choice: A set of short identifiers. The values of the identifiers in the set represent the choices made by the learner. The set may contain zero or more short identifiers. Examples: If a single choice was allowed, the set would contain a single identifier, e.g., "alpha." If a combination of choices was allowed, the set would contain multiple identifiers, e.g., "alpha," "bravo," and "delta," the order of which is insignificant.

(c) fill_in: An array of localized strings.

(d) long_fill_in: A localized string.

(e) likert: A short identifier. The value of the identifier represents the choice made by the learner.

(f) matching: A bag that contains zero or more records. Each record contains a source and a target that are represented by short identifiers. Each record represents a match made by the learner.

(g) performance: An array of responses in the order in which they were provided by the learner in response to the interaction. Each response consists of a step name (a short identifier) and either a single literal value (a character string) or a number. The step names and types of the responses shall match those provided in the correct_responses for the interaction (see 6.1.9.5), but the responses in the learner_response may be in a different order. Because a learner may perform the same step more than once, the step names of the responses may not be unique (i.e., a step name may appear more than once with the same value or with a different value).

Example: If the performance involves setting several valves to specific positions, the learner may adjust the position of the same valve more than once in the course of the performance. The name-value pairs for the response might be "valve 1:open, valve 2:closed, valve 1:closed." Notes: (1)The SPM for performance for learner_response is twice the size of the SPM for performance for correct_responses (see 6.1.9.5) to allow the recording of extra steps, as in the example above. (2) The syntax of the name-value pairs is not specified by this Standard.

(h) sequencing: An array of zero or more short identifiers. The sequence determined by the learner is represented by the order of the elements in the array. Each short identifier identifies one element that was available to be sequenced.

(i) numeric: A real number.

(j) other: A string defined by the specific "other" interaction type (see 6.1.9.2). The content of this string is not defined by this Standard.

Subclauses 6.2.5 and 6.2.9 define localized_string_type and short_identifier_type, respectively.

NOTES

1.The learner_response data element is a structured mechanism for identifying the exact learner response relating to each of the types of interactions described in 6.1.9.2. The determination of correctness is an implementation-defined feature of the content object.

2.The type of the interaction, as defined in 6.1.9.2, has to be known to select the appropriate variant.

6.1.9.8 Result

(1) Synopsis

```
result :
    choice
    (

        state( result_state, numeric ),
    )
    of
    (

        result_state :
            state( correct, incorrect,
            unanticipated, neutral ),
            numeric :
            real(10,7),

    ),
```

(2) Description

The value of this data element is a judgment of the correctness of the learner response. This data element shall have one of the following permissible values:

(a) correct: The learner response was correct.

(b) incorrect: The learner response was incorrect.

(c) unanticipated: The learner response was not expected.

(d) neutral: The learner response was neither correct nor incorrect.

(e) numeric: A real number.

NOTES

1.This Standard does not specify where or how the value of result is determined.

2.The numeric value real(10,7) is included to provide the capability of reporting a numeric estimate of the correctness of the learner response. This Standard does not specify how correctness is represented in the numeric value.

6.1.9.9 Latency

(1)Synopsis

latency :

timeinterval(second,10,2),

(2)Description

The value of this data element is the time elapsed between the time the interaction was made available to the learner for response and the time of the first response.

A string binding conforming to ISO 8601:2000 may be used to communicate time interval values (see Annex C).

NOTE：The latency information is not available for an interaction if the learner did not respond. The latency is, in effect, the time difference between the time_stamp (see 6.1.9.4) of the interaction and the time of the first response. If several interactions have the same time_stamp because they became available for response at the same time, the latency recorded for each interaction can be used to determine the order in which the learner responded to these interactions.

6.1.9.10 Description

(1)Synopsis

description :

localized_string_type(250),

// the parameter value is the SPM

(2)Description

The value of this data element is a brief informative description of the interaction. Subclause 6.2.5 defines localized_string_type.

6.1.10 Launch data

(1) Synopsis

launch_data :

characterstring(iso-10646-1),

// SPM: 4000 characters

(2) Description

The value of this data element provides data specific to a content object that the content object can use for initialization. The value of this data element is not specified.

NOTE：The allowable values for this data element are defined by the implementer of the content object. Typically, the documentation for the content object would specify what data can or has to be provided.

6.1.11 Learner ID

(1) Synopsis

```
learner_id :
    long_identifier_type,
```

(2) Description

The value of this data element identifies the learner on behalf of whom this content object instance was launched. The label shall be unique at least within the scope of the content object.

Subclause 6.2.6 defines long_identifier_type.

NOTE：This Standard does not specify how learner IDs are created, assigned, or resolved.

6.1.12 Learner name

(1) Synopsis

```
learner_name :
    localized_string_type(250),
    // the parameter value is the SPM
```

(2) Description

The value of this data element is the name of the learner.

Subclause 6.2.5 defines localized_string_type.

NOTE：This Standard does not specify how learner names are created, assigned, or resolved.

6.1.13 Learner preference data

(1) Synopsis

```
learner_preference_data :
    learner_preference_type,

type learner_preference_type =

    record

    (

        audio_level :

            real(10,7) range(0..*),
```

language :

language_type,

delivery_speed :

real(10,7) range(0..*),

audio_captioning :

state(off, no_change, on),

),

(2) Description

The values of this data element specify learner preferences associated with the learner's use of the content object.

The components of learner_preference_data are defined in 6.1.13.1 – 6.1.13.4.

NOTE：This Standard does not specify whether the content object, the RTS, or both have the ability to set or interpret learner preferences.

6.1.13.1 Audio level

(1) Synopsis

audio_level :

real(10,7) range(0..*),

(2) Description

The value of this data element is a multiplier value that specifies an intended change in perceived audio level relative to an implementation-specific reference level with the value 1 meaning "no change." For example, the value 0 specifies infinite attenuation, the value 0.5 specifies an attenuation of 10 decibels, and the value 2 specifies an amplification of 10 decibels.

NOTE：The multiplier value is not intended to be applied to the effect of previous changes communicated through this data element, but rather to the same implementation-specific reference level.

6.1.13.2 Language

(1) Synopsis

language :

language_type,

(2) Description

The value of this data element is the learner's preferred language for a content object with multilingual capability.

Subclause 6.2.4 defines language_type.

6.1.13.3 Delivery speed

(1) Synopsis

delivery_speed :
real(10,7) range(0..*),

(2) Description

The value of this data element is a multiplier that specifies the learner's preferred relative speed of content delivery expressed as a change in speed relative to an implementation-specific reference speed. For example, the value 2 is twice as fast as the reference speed and the value 0.5 is one half the reference speed.

NOTES

1.A value of 0 indicates that delivery is stopped.

2.The multiplier value is not intended to be applied to the effect of previous changes communicated through this data element, but rather to the same implementation-specific reference speed.

6.1.13.4 Audio captioning

(1) Synopsis

audio_captioning :
state(off, no_change, on),

(2) Description

The value of this data element specifies whether captioning text corresponding to audio is displayed. This data element shall have one of the following permissible values:

(a) off: Captioning is off, and text corresponding to audio is not displayed.

(b) no_change: The current captioning setting.

(c) on: Captioning is on, and text corresponding to audio is displayed.

6.1.14 Lesson status

(1) Synopsis

lesson_status :
state(passed, failed, completed, incomplete, browsed,
not_attempted),

(2) Description

This data element is included for backward compatibility with legacy implementations. The data elements completion_status and success_status should be used (see 6.1.3 and 6.1.24, respectively).

The value of this data element indicates whether the learner has attempted, completed, passed, failed, or browsed the associated content object. This data element shall have one of the following permissible values:

(a) passed: The learner has satisfied the requirements to pass the content object.
(b) failed: The learner has not satisfied the requirements to pass the content object.
(c) completed: The learner has satisfied the requirements to complete the content object.
(d) incomplete: The learner has not satisfied the requirements to complete the content object.
(e) browsed: The learner has accessed the content object with a mode of browse or elected to browse while in the content object after a normal launch.
(f) not_attempted: The learner has not accessed the content object, or the learner previously has accessed the content object but has experienced so little of it that it is considered to be not attempted.

NOTES

1. This Standard does not specify how to determine lesson_status. It may be reported by a content object, determined by an RTS by comparing scores to mastery scores, determined on the basis of objectives set by an outside agent (e.g., an instructor), or by some other means.
2. The completion_status and success_status data elements should be used because the lesson_status data element may exist only in legacy implementations.

6.1.15 Location

(1) Synopsis

location :
characterstring(iso-10646-1),
// SPM: 1000 characters

(2) Description

The value of this data element is a location in the content object. The value and its meaning are defined by the content object and are not specified by this Standard. This Standard does not specify how an implementation shall represent that there is no value for location.

NOTES

1. Depending on the implementation, the absence of a value for location could be represented as an empty string, a null element, or the absence of the data element.
2. If a content object communicates a location on exit, this data element provides support for a mechanism that lets the learner return to the content object at the same place he or she left it. This data element can identify the learner's exit point with a value that is meaningful to the content object only, and that location information can be used by the content object as an entry point the next time

the learner enters the content object. This data element also can be used by the content object to communicate its location to the RTS on an ongoing basis. Example: An RTS may be able to use this information to create bookmarks or to synchronize reference materials or annotations with the location reported by the content.

6.1.16 Max time allowed

(1) Synopsis

max_time_allowed :

timeinterval(second,10,2),

(2) Description

The value of this data element is the amount of accumulated time the learner is allowed to use a content object in the learner attempt. (See 6.1.26 for the content object's expected response to exceeding the limit.)

A string binding conforming to ISO 8601:2000 may be used to communicate time interval values (see Annex C).

NOTE：The learner attempt begins with the beginning of the first learner session and continues until the activity terminates.

6.1.17 Mode

(1) Synopsis

mode :

state(browse, normal, review),

(2) Description

The value of this data element identifies one of three possible modes in which a content object may be presented to a learner. This data element shall have one of the following permissible values:

(a) browse: The content object is presented without the intent of recording any information about the current learner session.

(b) normal: The content object is presented with the intent of recording information about the current learner session.

(c) review: The content object has previously recorded information about the learner attempt and is presented without the intent of updating this information with data from the current learner session. Note: The learner attempt begins with the beginning of the first learner session and continues until the activity terminates.

6.1.18 Objectives

(1) Synopsis

objectives :

```
            set of objective_type,
            // SPM: 100 objective_type records in the bag

type objective_type =
    record
    (

        id :
            long_identifier_type,
            score :
            score_type,
            status :
            state( passed, completed, failed, incomplete, browsed,
              not_attempted ),
            progress_measure :
            progress_measure_type,
            completion_status :
            completion_status_type,
            success_status :
            success_status_type,
            description :
            localized_string_type(250),
            // the parameter value is the SPM

    ),
```

(2) Description

The values of this data element specify learning or performance objectives associated with a content object. An instance of an objective_type record shall include an objective identifier (see 6.1.18.1); all other components are optional.

The components of objective_type are defined in 6.1.18.1 – 6.1.18.7.

NOTES

1. Information about objectives may come from a content object, from an RTS, or from some other source.
2. This Standard does not define any relationship between objectives and the content object's completion_status, lesson_status, score, or success_status (see 6.1.3, 6.1.14, 6.1.22, and 6.1.24, respectively).

6.1.18.1 ID

(1) Synopsis

id :

long_identifier_type,

(2) Description

The value of this data element is a label for the objective. This label shall be unique at least within the scope of the content object.

Subclause 6.2.6 defines long_identifier_type.

NOTE：This Standard does not specify how IDs are created, assigned, or resolved.

6.1.18.2 Score

(1) Synopsis

score :

score_type,

(2) Description

The value of this data element is the score achieved by the learner for the objective.

Subclause 6.2.8 defines score_type.

NOTE：This Standard does not specify how the value of score is created or assigned.

6.1.18.3 Status

(1) Synopsis

status :

state(passed, completed, failed,

incomplete, browsed,

not_attempted),

(2) Description

This data element is included for backward compatibility with legacy implementations. The data elements completion_status and success_status should be used (see 6.1.18.5 and 6.1.18.6, respectively).

The value of this data element indicates whether the learner has engaged with that portion of the content object related to the objective and, if so, whether the learner has demonstrated mastery of the objective. This data element shall have one of the following permissible values:

(a) passed: The objective was passed.

(b) completed: All parts of the content object related to the objective were accessed. The objective may or may not have been passed.

(c) failed: The objective was failed.

(d) incomplete: Not all parts of the content object related to the objective were accessed.

(e) not_attempted: No part of the content object related to the objective was accessed.

(f) browsed: No specific status information for the objective is available because the content object related to the objective was launched with a mode of browse (see 6.1.17).

NOTES

1.This Standard does not specify how to determine status. Status may be provided by the content object, by an RTS, or by some other means.

2.The completion_status and success_status data elements should be used because the status data element may exist only in legacy implementations.

6.1.18.4 Progress measure

(1) Synopsis

progress_measure :
progress_measure_type,

(2) Description

The value of this data element is a measure of the progress the learner has made toward completing the objective.

Subclause 6.2.7 defines progress_measure_type.

NOTE：This Standard does not specify how to determine the value of progress_measure.

6.1.18.5 Completion status

(1) Synopsis

completion_status :
completion_status_type,

(2) Description

The value of this data element indicates whether the learner has completed the objective.

Subclause 6.2.2 defines completion_status_type.

NOTE：This Standard does not specify how to determine completion_status. It may be reported by a content object, determined by an RTS, determined on the basis of objectives set by an outside agent (e.g., an instructor), or by some other means.

6.1.18.6 Success status

(1) Synopsis

success_status :
success_status_type,

(2) Description

The value of this data element indicates whether the learner has mastered the objective.

Subclause 6.2.10 defines success_status_type.

NOTE：This Standard does not specify how to determine success_status. It may be reported by a content object, determined by an RTS, determined on the basis of objectives set by an outside agent (e.g., an instructor), or by some other means.

6.1.18.7 Description

(1) Synopsis

```
description :
    localized_string_type(250),
    // the parameter value is the SPM
```

(2) Description

The value of this data element is a brief informative description of the objective.

Subclause 6.2.5 defines localized_string_type.

6.1.19 Progress measure

(1) Synopsis

```
progress_measure :
    progress_measure_type,
```

(2) Description

The value of this data element is a measure of the progress the learner has made toward completing the content object.

Subclause 6.2.7 defines progress_measure_type.

NOTES

1. This Standard does not specify an exact relationship between completion_status (see 6.1.3) and values for progress_measure other than 0 or 1. Any value between 0 and 1 typically corresponds to a completion_status value of incomplete, unless the value is equal to or above a defined completion_threshold (see 6.1.4); in which case, the value typically corresponds to a completion_status value of completed.
2. This Standard does not specify how to determine the value of progress_measure.

6.1.20 Raw passing score

(1) Synopsis

```
raw_passing_score :
    real(10,7),
```

(2)Description

The value of this data element is the raw passing score for a content object. The scale is not defined. This data element is included for backward compatibility with legacy implementations. The data element scaled_passing_score should be used (see 6.1.21).

NOTE：The scaled_passing_score data element should be used because the raw_passing_score data element may exist only in legacy implementations.

6.1.21 Scaled passing score

(1)Synopsis

scaled_passing_score :
real(10,7) range(-1..1),

(2)Description

The value of this data element is the scaled passing score for a content object. The value of this data element is scaled to fit the range –1 to 1 inclusive.

NOTES

1.If a scaled_passing_score is defined for the use of a content object, this is a statement that the requirements associated with the use of that content object are achieved by obtaining a score (see 6.1.22) greater than or equal to the scaled_passing_score. For example, if the scaled_passing_score for a content object is 0.85 and a learner achieves a scaled score of 0.90, a success_status of passed may be assigned to that content object for that learner (see 6.1.24). However, this Standard does not specify or require that an RTS, content object, or any other system component interpret or take action in response to a scaled_passing_score.

2. A scaled score range of –1 to +1 is used to allow a content developer to more easily assign a penalty for an incorrect choice.

6.1.22 Score

(1) Synopsis

score :
score_type,

(2) Description

The value of this data element is the learner's score for the content object.

Subclause 6.2.8 defines score_type.

6.1.23 Session time

(1) Synopsis

session_time :
timeinterval(second,10,2),

(2) Description

The value of this data element is the amount of time that the learner has spent in the current learner session for this content object. If no learner session is in progress, the session time is the time the learner spent in the last learner session for this content object.

A string binding conforming to ISO 8601:2000 may be used to communicate time interval values (see Annex C).

NOTES

1. This Standard does not specify how to determine the value of session_time or its accuracy.
2. The value for session_time may be evaluated one or more times during a learner session. The value of total_time (see 6.1.27) is not updated until after the learner session has ended.
3. If a learner session is in progress, the actual duration of the learner attempt is the total_time (see 6.1.27) plus the current session_time.

6.1.24 Success status

(1) Synopsis

success_status :
success_status_type,

(2) Description

The value of this data element indicates whether the learner has mastered the content object. Subclause 6.2.10 defines success_status_type.

NOTE：This Standard does not specify how to determine success_status. It may be reported by a content object, determined by an RTS by comparing scores to mastery scores, determined on the basis of objectives set by an outside agent (e.g., an instructor), or by some other means.

6.1.25 Suspend data

(1) Synopsis

suspend_data :
characterstring(iso-10646-1),
// SPM: 4000 characters

(2) Description

The value of this data element provides information that may be created by a content object as a result of a learner accessing or interacting with that content object. The format of the content of this data element is

unspecified.

NOTE：The intent is for the content object to store data for later use in the current learner session or a subsequent learner session between the content object and the same learner.

6.1.26 Time limit action

(1) Synopsis

time_limit_action :

state(exit_message, continue_message, exit_no_message,
continue_no_message),

(2) Description

The value of this data element indicates what the content object should do when max_time_allowed is exceeded (see 6.1.16). This data element shall have one of the following permissible values:

(a) exit_message: The learner should be forced to exit the content object. The content object should provide a message to the learner that indicates that the maximum time allowed for the learner attempt was exceeded.

(b) continue_message: The learner should be allowed to continue in the content object. The content object should provide a message to the learner that indicates that the maximum time allowed for the learner attempt was exceeded.

(c) exit_no_message: The learner should be forced to exit the content object with no message.

(d) continue_no_message: Although the learner has exceeded the maximum time allowed for the learner attempt, the learner should not be given a message and should not be forced to exit the content object.

NOTES

1. When a message is presented to the learner, the content object defines the content and form of the message.
2. This Standard does not specify how the content object forces the learner to exit the content object.

6.1.27 Total time

(1) Synopsis

total_time :

timeinterval(second,10,2),

(2) Description

The value of this data element is the sum of all of the learner's learner session times (see 6.1.23) accumulated in the current learner attempt before the current learner session. The value of total_time shall not be updated while a learner session is in progress.

A string binding conforming to ISO 8601:2000 may be used to communicate time interval values (see Annex C).

NOTE：The learner attempt begins with the beginning of the first learner session and continues until the activity terminates.

6.2 Auxiliary data types

The following data types are used in conjunction with the data elements described in 6.1.

6.2.1 Comment type

(1) Synopsis

```
type comment_type =
    record
    (

        comment :
            localized_string_type(4000),
            // the parameter value is the SPM
            location :
            characterstring(iso-10646-1),
            // SPM: 1000 characters
            time_stamp :
            date_time_type,
            ),
```

(2) Description

This data type describes textual input. Instances of this data type shall include a comment (see 6.2.1.1). The components of the comment_type are defined in 6.2.1.1 – 6.2.1.3.

6.2.1.1 Comment

(1) Synopsis

```
comment :
    localized_string_type(4000),
    // the parameter value is the SPM
```

(2) Description

This data element shall describe comments or annotations associated with a content object.

Subclause 6.2.5 defines.

NOTE：This Standard does not define a structure or format for the content of the localized string.

6.2.1.2 Location

(1) Synopsis

location :
characterstring(iso-10646-1),
// SPM: 1000 characters

(2) Description

This data element is the point in the content object at which the comment applies. If no value is specified for location, the comment is applicable to the entire content object. This Standard does not specify how an implementation shall represent that no value exists for location.

NOTES

1.Depending on the implementation, the absence of a value for location could be represented as an empty string, a null element, or the absence of the data element.

2. This Standard does not specify how an implementation defines a location in a content object.

6.2.1.3 Time stamp

(1) Synopsis

time_stamp :
date_time_type,

(2) Description

This data element is the point in time at which the comment was created or most recently changed. Subclause 6.2.3 defines date_time_type.

6.2.2 Completion status type

(1) Synopsis

type completion_status_type =
state(completed, incomplete, not_attempted, unknown),

(2) Description

This data type indicates whether the learner has completed a content object or an objective. This data type shall have one of the following permissible values:

(a) completed: The learner has experienced enough of the content object or objective to consider it completed.

(b) incomplete: The learner has not experienced enough of the content object or objective to consider it completed.

(c) not_attempted: The learner is considered not to have used the content object or objective in

any significant way. Note: The learner has not accessed the content object or objective, or the learner previously has accessed it but has experienced so little of it that it is considered to be not attempted.

(d) unknown: No assertion is made.

6.2.3 Data time type

(1)Synopsis

```
type date_time_type =
    time(second,10,0),
```

(2)Description

This data type represents a point in time. This data type shall have a required precision of 1 second and an optional precision of 0.01 seconds.

Implementations of this data type shall include distinct representations for points in time in the range 197001-01 00:00:00 through 2037-12-31 23:59:59, not including leap seconds, with a required precision of 1 second and an optional precision of 0.01 seconds. Implementations may include distinct representations for values beyond the required date and time range.

A string binding conforming to ISO 8601:2000 may be used to communicate date and time values (see Annex C).

NOTES

1. This Standard does not specify how to translate times expressed with precisions of hundredths of a second to times expressed with precisions of seconds, which may be done by rounding, truncation, or another method.
2. Conforming implementations are permitted, but not required, to support the representation of leap seconds.

6.2.4 Language type

(1) Synopsis

```
type language_type =
    characterstring(iso-646),
    // SPM: 250 characters
```

(2) Description

The format of this data type is a character string consisting of a required language code followed by multiple, optional, hyphen-prefixed subcodes (see examples below).

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

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

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

(c) The one-letter code “i” is reserved and used as a prefix for registrations defined by the

Internet Assigned Numbers Authority (IANA).

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

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

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

(b) Subcodes from three to eight letters are registered with IANA.

Rules for additional subcodes are unspecified.

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

Examples

“en-GB”

“de”

“fr-CA”

“it”

“grc” (Ancient Greek, until 1453)

“en-US-philadelphia”

“eng-GB-cockney”

“map-PG-buin”(Austronesian - Papua New Guinea Buin)

“gem-US-pennsylvania”

“i-bnn” (IANA Bunun)

6.2.5 Localized string type

(1) Synopsis

```
type localized_string_type(length) =
    record
    (

        language :
            language_type,
            string :
            characterstring(iso-10646-1),
            // SPM: the length parameter
            ),
```

(2) Description

This data type consists of a language specification for a string and the string itself.

The components of the localized_string_type are defined in 6.2.5.1 and 6.2.5.2.

Examples

The following are three examples of localized strings: “Information Technology” in French, “localization” in British English, and “xxx” in Japanese hiragana.

(“fr”, “Technologies de l’information”)

(“en-GB”, “localisation”)

(“jp-JP-jisx208”, “xxx”)

6.2.5.1 Language

(1)Synopsis

language :

language_type,

(2)Description

This data element specifies the language of the localized string.

Subclause 6.2.4 defines language_type.

6.2.5.2 String

(1) Synopsis

string :

characterstring(iso-10646-1),

// SPM: the length parameter

(2) Description

This data element contains the text of the localized string.

6.2.6 Long identifier type

(1) Synopsis

type long_identifier_type =

characterstring(iso-10646-1),

// SPM: 4000 characters

(2) Description

This data type is an identifier (a label) associated with an object that is intended to be unique within the context of usage of the object. The character string shall conform to the syntax for Uniform Resource Identifiers (URIs) as defined by RFC 2396.

NOTE：This Standard recommends that the URI be a globally unique identifier in the form of a Uniform Resource Name (URN) (see RFC 2141 [B4]).

6.2.7 Progress measure type

(1)Synopsis

```
type progress_measure_type :
    real(10,7) range(0..1),
```

(2)Description

This data type is a measure of the progress the learner has made toward completing a content object or an objective. A value of 0 corresponds to a completion_status_type value of not_attempted (see 6.2.2). A value of 1 corresponds to a completion_status_type value of completed.

6.2.8 Score type

(1) Synopsis

```
type score_type =
    record
    (
        raw :
            real(10,7),
            min :
            real(10,7),
            max :
            real(10,7),
            scaled :
            real(10,7) range(-1..1),
    ),
```

(2) Description

This data type describes scoring information.

The components of the score_typeare defined in 6.2.8.1 – 6.2.8.4.

6.2.8.1 Raw

(1) Synopsis

```
raw :
    real(10,7),
```

(2) Description

This data element is a number that reflects the performance of the learner relative to the range bounded by the values of min and max.

NOTE：A raw score is not necessarily an unprocessed score. Example: An unprocessed score might be converted to a percentage score as a decimal value. That is, the learner achieved a score of 3 out of 4 possible, which is converted to a raw value of 0.75 with min (see 6.2.8.2) equal to 0

and max(see 6.2.8.3) equal to 1.

6.2.8.2 Min

(1) Synopsis

min :

real(10,7),

(2) Description

This data element is the minimum value in the range for the raw score (see 6.2.8.1).

6.2.8.3 Max

(1) Synopsis

max :

real(10,7),

(2) Description

This data element is the maximum value in the range for the raw score (see 6.2.8.1).

6.2.8.4 Scaled

(1) Synopsis

scaled :

real(10,7) range(-1..1),

(2) Description

This data element is a number that reflects the performance of the learner. The value of the data element is scaled to fit the range –1 to 1, inclusive.

NOTE：A scaled score range of –1 to +1 is used to allow a content developer to more easily assign a penalty for an incorrect choice, such as in a flight simulation system where the learner's choice would have resulted in the loss of the aircraft and all aboard.

6.2.9 Short identifier type

(1) Synopsis

type short_identifier_type =

characterstring(iso-10646-1),

// SPM: 250 characters

(2) Description

This data type is an identifier (a label). The character string shall conform to the syntax for URIs as defined by RFC 2396.

6.2.10 Success status type

(1) Synopsis

type success_status_type =

state(passed, failed, unknown),

(2) Description

This data type indicates whether the learner has mastered a content object or an objective. This data type shall have one of the following permissible values:

(a) passed: The learner has passed the content object or objective.

(b) failed: The learner has failed the content object or objective.

(c) unknown: No assertion is made.

Annex A

Bibliography

(informative)

[B1] AICC CMI001, CMI Guidelines for Interoperability, Version 3.5, April 2001.

[B2] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

[B3] IEEE Std 1484.11.2-2003, Standard for Learning Technology—ECMAScript Application Programming Interface for Content to Runtime Services Communication.

[B4] IETF RFC 2141, URN Syntax.

Annex B

Understanding the ISO/IEC 11404:1996 real and time interval data type definitions used in this Standard

(informative)

The real and time interval data types used in this Standard are discussed in B.1 and B.2.

B.1 Real data type

The declaration real(10,7) denotes a real data type with values that have precision to 10^{-7} (i.e., 0.0000001). For example, according to this type definition

— 5550.000001 and 5550.000002 are different values

— 5550.000000001 and 5550.0 may evaluate to the same value, because the difference of 0.000000001 is too small to be accounted for according to the precision requirement of the type definition

— 5550.0 and 5550.000000 are the same value

— 5550.0 and 5550 evaluate to the same value

B.2 Time interval data type

The declaration timeinterval(second,10,2) denotes that the value for the data element timeinterval represents elapsed time with a precision of 0.01 seconds.

This Standard does not require implementations to distinguish between, for example, time intervals of 2.000 seconds and 2.001 seconds, because the difference of 0.001 seconds is less than the precision requirement for this data type.

This Standard recommends that bindings use a string representation conforming to ISO 8601:2000 to communicate the time interval value. However, this Standard does not specify a binding, and different bindings are possible for a value of this data type.

For example, if a binding uses real numbers to represent seconds

— A duration of exactly 1 hour can be expressed with the real value 3600.0

— A duration of 2.5 seconds can be expressed with the real value 2.5

— A duration of 1 hour and 30 minutes can be expressed with the real value 5800.0

If a binding uses ISO 8601:2000

— A duration of exactly 1 hour can be expressed with the string “PT1H”

— A duration of 2.5 seconds can be expressed with the string “PT2.5S”

— A duration of 1 hour and 30 minutes can be expressed with the string “PT1H30M”

The format for the string representations above is defined by the following pattern:

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

where

y is the number of years (integer, ≥ 0, not restricted)

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 (real or integer, ≥ 0, not restricted, e.g., > 59 is acceptable)

The character literal designators “P”, “Y”, “M”, “D”, “T”, “H”, “M”, and “S” have to appear if the corresponding nonzero value is present.

Annex C

ISO 8601:2000 representation of the date time type

(informative)

A string representation conforming to ISO 8601:2000 may be used to communicate the values of the date_time_type (see 6.2.3). This Standard does not specify a binding, and other bindings are possible.

For example, using a string representation conforming to ISO 8601:2000, the point in time July 16, 1997, 30.17 seconds past 7:20 PM with a time offset of 1 hour with respect to UTC, can be expressed with the string

"1997-07-16T19:20:30.17+01:00"

where the format is defined by the following pattern:

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 month (01 through 31, depending on value of month and year)

hh is the two digits of hour (00 through 23) (AM/PM NOT allowed)

mm is the two digits of minute (00 through 59)

ss is the two digits of second (00 through 59)

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

TZD is the time zone designator ("Z" for UTC or +hh:mm or -hh:mm)

At least the four-digit year must be present. If additional parts of the value of the date_time_type are included, the character literals "-", "T", ":", and "." are parts of the character lexical representation for the value.

If the time portion is present, but the time zone designator is not present, the time zone is unspecified, and the time is interpreted as "local time."

教材物件溝通資料模型之可延伸標示語言架構繫結 —
參考資料、爭議事項、英中名詞對照

1. 參考資料

IEEE Std 1484.11.1-2004, IEEE Standard for Learning Technology—Data Model for Content Object Communication.2

W3C Recommendation (28 October 2004), XML Schema Part 1: Structures, Second Edition.3

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

2. 爭議事項

由於本文件是規範資料模型的繫結方式，爲保持與國際標準的相容性，在研譯過程中皆保留原始標準(IEEE 1484.11.3)的定義，並沒有發現爭議之處。

3. 英中名詞對照

- A -

array	陣列
attribute	屬性
audio captioning	音訊字幕

- B -

bag	紀錄袋
binding	繫結

- C -

cardinality	基數
character	字元
character string	字元串
content	教材
content object communication data (COCD)	教材物件溝通資料
convention	規約
credit	學分

- D -

data-model	資料模型

- E -

element	元素
enumerated type	列舉型式

Extensible Markup Language(XML)	可擴展標示語言

- F -

fragment	片段

- G -

- H -

- I -

identifier	識別符
instance	實例
interaction	互動

- J -

- K -

- L -

latency	潛伏時間
literal	文字

- M -

max time allowed	最大允許時間

- N -

- O -

- P -

prefix	冠碼

- Q–

- R -

raw passing score	原始及格分數
root element	根元素

- S -

scaled passing score	標度及格分數

schema	架構
session time	連線時間
SPM(smallest permitted maximum)	最小允許上限值
suspend data	懸置資料

- T -

time stamp	時戳

- U -

- V -

- W -

W3C (World Wide Web Consortium)	全球資訊網聯盟

- X -

XSD(XML Schema definition)	XSD (XML 架構定義)

- Y -

- Z -

4. 中英名詞對照表

陣列	array
屬性	attribute
音訊字幕	audio captioning
紀錄袋	bag
繫結	binding
基數	cardinality
字元	character
字元串	character string
教材	content
教材物件溝通資料	content object communication data (COCD)
規約	convention
學分	credit
資料模型	data-model
元素	element
列舉型式	enumerated type

可擴展標示語言	Extensible Markup Language(XML)
片段	fragment
識別符	identifier
實例	instance
互動	interaction
潛伏時間	latency
文字	literal
最大允許時間	max time allowed
冠碼	prefix
原始及格分數	raw passing score
根元素	root element
標度及格分數	scaled passing score
架構	schema
連線時間	session time
最小允許上限值	SPM(smallest permitted maximum)
懸置資料	suspend data
時戳	time stamp
全球資訊網聯盟	W3C (World Wide Web Consortium)
XSD (XML 架構定義)	XSD(XML Schema definition)