

CNS 國家標準草案：教材物件溝通資料模型之可
延伸標示語言架構繫結

**(IEEE 1484.11.3—IEEE Standard for Learning
Technology—Extensible Markup Language (XML) Schema
Binding for Data Model for Content Object
Communication)**

中央研究院·數位典藏國家型科技計畫後設資料工作組

與

財團法人資訊工業策進會數位教育研究所

研譯

經濟部標準檢驗局 96 年度 委辦計畫

規劃與建置數位內容與數位生活應用之技術標準環境

(案號：1D15960125-20)

目錄

1.適用範圍	1
2.用語釋義	2
3.參考標準	4
4.符合性	4
5.XML 繫結	6
附錄 A 參考文獻	7
附錄 B XSD 規範	8
附錄 C COCD XML 實例範例	58
附錄 D XSD 註釋	75
附錄 E 網路可使用之 XSD 檔案與範例實例	142
英中名詞對照表	143

1.適用範圍

1.1 範圍

本標準係為 IEEE 1484.11.1TM-2004.標準中全球資訊網聯盟(World Wide Web Consortium,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。

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. 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)。

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 的互動 HTML 網頁。

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)(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。

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.

3.1.1 content 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 Acronyms and abbreviations

COCD	content object communication data
SPM	smallest permitted maximum
W3C	World Wide Web Consortium
XML	Extensible Markup Language

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|版本為證；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 Std 1484.11.1-2004, IEEE Standard for Learning Technology—Data Model for Content Object Communication.

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

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，此類實例被稱為「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”，且應存於使用第 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 實例完成為一份 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 標準資料模型的需求(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（係為 http://ltsc.ieee.org/xsd/1484_11_3）。

附錄 B 中的 XSD 符合 XML 架構的第 1 部和第 2 部(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, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

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

Annex A

(informative)

Bibliography

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

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

附錄 B

(規定)

XSD規範

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.

```
<?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:element>

```

```

    </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:documentation>

</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
    </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:complexType>

    </xs:element>

  </xs:sequence>

</xs:group>
```

```

    </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: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:restriction>

</xs:simpleType>

```

```

        <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">
            Implements learner_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:all>
</xs:complexType>

```

```
</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:element>
```

```

    </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: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:complexType>
```

```

</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>
```

圖 B.1—符合的 XSD

Figure B.1—Conforming XSD

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.

```
<?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>
</cocd>
```

```
</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>
  <identifiant>urn:ostyn.com:LIQ2003090412345</identifiant>
  <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>
  <identifiant>urn:ostyn.com:MAQ2003090412345</identifiant>
  <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>
  </performancePattern>
</correctResponses>
```

```
</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>
    </stepSequence>
  </correctResponses>
</interaction>

```

```
<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>

</interaction>
```

```
</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>

```

圖C.1 COCD XML實例範例

Figure C.1—An example COCD XML instance

附錄 D

(參考)

XSD 註釋

D.1 一般

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(參照附錄 B)及 XML 實例的範例(參照附錄 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，也不打算為 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 1 部和第 2 部。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` 代表類型，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類型宣告：首先是元件然後是屬性，群組，簡單類型，和複雜類型。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

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

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。對於類似元件的收集則使用 `xs:sequence`，因為此係唯一能同時包含多個元件的方法，即使 `xs:sequence` 隱含複雜類型的教材必須看似一次序列表。下述範例為一典型紀錄袋/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，其為集合模型的特性。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 2 部分的規定，“此規格不為一些資料類型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類型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` 都被定義成類型定義的一部份，XSD 定義一個屬性名為 `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，因為一個 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”，將為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為次序的或無序的。

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中來實作收集中的獨特性，其定義為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 正確回應和 6.1.9.7 學習者對多重選擇互動回應中。
- (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，對互動類型多重選擇集合中正確回應之選擇集合中，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，因為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

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 類型設置 `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為 `spm`。此屬性被加給數個元件和類型，這些元件和類型是定義在 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，也不允許此屬性為必須保留簡單類型的字串類型定義，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，因為 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 `spm` attribute.

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"`，將為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

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與元件關聯的 SPM，當 XSD 沒有提供 SPM 值作為帶有一固定值之 `spm` 屬性時，或是實作可能會在 XSD 沒有提供收集類型屬性(此收集類型屬性帶有指出收集類型的固定值)時使用收集類型。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。(為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) 收集類型(`collectionType`)：此名稱代表用`xs:sequence`編碼的收集類型。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。為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。為了發現SPM或XSD定義中對元件的收集類型，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`或收集類型(`spm` or `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 教材物件溝通6.1之編碼

元件於資料模型中被載明為 content_object_communication : record ，在XSD中利用cocdType 類型之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。

每個存在於複雜類型之 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:all>
</xs:complexType>
```

```

...
<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` 類型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 學習者評論6.1.1之編碼

此資料模型於XSD中被實作成`commentsFromLearner`元件，此元件可以出現次數為“0”次 或是“多次”。`commentFromLearner`元件都為`commentType`類型，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 學習管理平台評論6.1.2之編碼

此資料模型於XSD中被實作成`commentsFromLMS`元件，此元件可以出現次數為“0”次 或是“多次”。`commentFromLMS`元件都是`commentType`類型，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 1484.11.12004.

D.3.3 完成狀態6.1.3之編碼

此資料模型於XSD中被實作成Completion元件，其類型為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 完成門檻6.1.4之編碼

此資料模型於XSD中被實作成completionThreshold元件，其類型為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 學分6.1.5之編碼

此資料模型於XSD中被實作成credit元件，其類型為帶有標記值之列舉類型(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 資料模型版本6.1.6之編碼

此資料模型於XSD中被實作成`dataModelVersion`元件，其類型為`literalString250Type`且為一個自訂類型，主要的功能在於當其值為空白時避免突發性的修改狀況發生。此類型為一個值為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 進入6.1.7之編碼

此資料模型於XSD中被實作成`entry`元件，其類型為帶有標記值之列舉類型(`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 之實例中可以被視為當作一個空元件(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 退出 6.1.8 之編碼

此資料模型於XSD中被實作成exit元件，其類型為帶有標記值之列舉類型(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 之實例中可以被視為當作一個空元件(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 互動6.1.9之編碼

此資料模型於 XSD 中被實作成 Interactions 元件，其類型為 interactionsType 之元件序列。此 interactionsType 類型被定義為收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為收集 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 互動類型之實作

此資料模型結構於 XSD 中被實作成公用類型 InteractionType，包括了必備元件 identifier、type、可選擇的元件 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序出現。correctResponses 元件和 learnerResponse 元件被定義為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:all>
  </xs:complexType>
```

```

<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 ID 6.1.9.1之編碼

此資料模型於XSD中被實作成identifier元件，其類型為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 類型6.1.9.2之編碼

此資料模型於XSD中被實作成type元件，其類型為interactionTypeType。

```
<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 互動類型類型

此資料模型結構於XSD中被實作成公用類型InteractionTypeType，其主要被定義為帶有標記值之列舉類型(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 目標ID 6.1.9.3之編碼

此資料模型於XSD中被實作成`ObjectiveIds`元件，其類型為`objectiveIdsType`。此模型包含“0”個至“多個”`longIdentifierType`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 `longIdentifierType`. A uniqueness constraint is applied.

`objectiveIdsType`公用類型被定義為收集`longIdentifierType`類型之`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 `longIdentifierType`.

D.3.9.6 時戳6.1.9.4之編碼

此資料模型於 XSD 中被實作成 `timestamp` 元件，其類型為 `dateTimeType`。

```
<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 正確回應6.1.9.5之編碼

此資料模型於 XSD 中被實作成 `correctResponse` 元件，其類型為 `correctResponsesType`。

```
<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 正確回應類型

此公用類型實作了不同的 `correct_responses` 之資料結構，並定義了不同類型的互動方式。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 XSD 中。COCD XML 實例的實作可選擇一個正確的群組與此互動類型配對。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(參照附錄 C)做為呈現此類型之真實應用案例。

```
<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 加權6.1.9.6之編碼

此資料模型於 XSD 中被實作成 weighting 元件，其類型為 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 學習者回應6.1.9.7之編碼

此資料模型於 XSD 中被實作成 learnerResponse 元件，其類型為 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 學習者回應類型

此公用類型，實作各種定義在資料模型中之學習者回應(learner_response)資料結構，用來對應各種互動類型。無論互動類型資料結構如何複雜，選項都實作成XSD中的公用群組。由COCD XML實例的實作，決定選擇符合互動類型的正確群組。參照COCD XML實例(參照附錄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 成果6.1.9.8之編碼

此資料模型於XSD中被實作成result元件，其類型為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 互動成果類型

此公用類型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 潛時6.1.9.9之編碼

此資料模型於 XSD 中被實作成 `latency` 元件，其類型為 `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 描述6.1.9.10之編碼

此資料模型於 XSD 中被實作成 `description` 元件，其類型為 `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

此資料模型於 XSD 中被實作成 `launchData` 元件，其類型為 `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 學習者ID6.1.11之編碼

此資料模型於 XSD 中被實作成 `learnerId` 元件，其類型為 `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 學習者名稱6.1.12編碼

此資料模型於 XSD 中被實作成 `learnerName` 元件，其類型為 `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 學習者偏好資料6.1.13之編碼

此資料模型於 XSD 中被實作成 `learnerPreferenceData` 元件，其類型為 `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公用類型實作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 音訊級別6.1.13.1之編碼

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類型為基礎，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 語言6.1.13.2之編碼

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。其類型為 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 傳遞速度6.1.13.3之編碼

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 類型為基礎，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 音訊字幕6.1.13.4之編碼

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。其類型為擁有一個標記值的列舉類型，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 課程狀態6.1.14之編碼

此資料模型於XSD中被實作成`lessonStatus`元件，其類型為`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 位置6.1.15之編碼

此資料模型於XSD中被實作成`location`元件，其類型為`literalString1000Type`，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 最大允許時間6.1.16之編碼

此資料模型於XSD中被實作成maxTimeAllowed元件，其類型為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 模式6.1.17之編碼

此資料模型於 XSD 中被實作成 mode 元件，為擁有一個標記值的列舉類型，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 目標6.1.18之編碼

此資料模型於 XSD 中被實作成 objective 元件序列，其類型為 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 不可被解釋為隱含任何特定次序。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>
```

公用類型的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 目標類型

此資料模型結構於 XSD 中被實作成 `objectiveType` 公用類型，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 進展測量6.1.19之編碼

此資料模型於 XSD 中被實作成 `progressMeasure` 元件，其類型為 `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 原始及格分數6.1.20之編碼

此資料模型於 XSD 中被實作成 `rawPassingScore` 元件，其類型為 `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 標度及格分數6.1.21之編碼

此資料模型於 XSD 中被實作成 `scaledPassingScore` 元件，其類型為 `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 分數6.1.22之編碼

此資料模型於 XSD 中被實作成 `score` 元件，其類型為 `scoreType`，該類型是由數個元件組成的複雜類型。

```
<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 交談時間6.1.23之編碼

此資料模型於 XSD 中被實作成 `sessionTime` 元件，其類型為 `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 成功狀態6.1.24之編碼

此資料模型於 XSD 中被實作成 `successStatus` 元件，其類型為 `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 懸置資料6.1.25之編碼

此資料模型於 XSD 中被實作成 `suspendData` 元件，其類型為 `literalString4000Type`，為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 時間限制動作6.1.26之編碼

此資料模型於 XSD 中被實作成 `timeLimitAction` 元件，其類型為擁有一個標記值的列舉類型，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:restriction>
  </xs:simpleType>
</xs:element>
```

```

        <xs:enumeration value="continue_no_message"/>
        <xs:enumeration value="exit_message"/>
        <xs:enumeration value="exit_no_message"/>
    </xs:restrictio
n>
    </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 總時間6.1.27之編碼

此資料模型於 XSD 中被實作成 `totalTime` 元件，其類型為 `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 輔助資料類型 6.2 之編碼

輔助資料類型之編碼在 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 評論類型 6.2.1 之實作

此資料模型類型於 XSD 中被實作成 commentType 公用類型，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 完成狀態類型 6.2.2 之實作

此資料模型類型於 XSD 中被實作成 completionStatusType 公用類型，定義為擁有一個標記值的列舉類型，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 資料時間類型 6.2.2 之實作

此資料模型類型於XSD中被實作成`dateTimeType`公用類型，是建立在XML架構的原始資料類型`dateTime`。 `dateTime`類型是符合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]]]]
```

括號中的數值為可選擇的。

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 語言類型 6.2.4 之實作

此資料模型類型於 XSD 中被實作成 `languageType` 公用類型，是建立在 XML 架構的衍生資料類型語言(`language`)。 `language` 類型是 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 定位字串類型6.2.5之實作

此資料模型類型於XSD中被實作成`localizedStringType`公用類型，建立在`literalStringType`公用類型的基礎上。此基本類型被用於避免當隱含空白時，字串值遭意外的修改。此 `localizedStringType`類型用來當作其他2種類型：`localizedString250Type`和 `localizedString4000Type`的抽象基本類型，此2種類型皆隱含對`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 長識別符類型 6.2.6 之實作

此資料模型類型於XSD中被實作成`longIdentifierType`公用類型，其建立於XML架構的原始資料類型`anyURI`基礎中，`anyURI`類型為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。因為此類型用在當簡單類型需求時，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 進展測量類型6.2.7之實作

此資料模型類型於XSD中被實作成progressMeasureType公用類型，建立在real7Type公用類型基礎上，也就是建立在XML架構允許的十進位資料類型基礎上。

```
<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 分數類型6.2.8之實作

此資料模型類型於XSD中被實作成scoreType公用類型，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 短識別符類型6.2.9之實作

此資料模型類型於XSD中被實作成shortIdentifierType公用類型，建立於XML架構anyURI原始資料類型中，anyURI類型為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。因為此類型將用在當簡單類型要求時，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 成功狀態類型6.2.10之實作

此資料模型類型於XSD中被實作成successStatusType公用類型，其定義為擁有一個標記值的列舉類型，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 其他文件資料類型之實作

其他文件資料類型之實作將在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)`之實作

此類型的解釋定義於 IEEE 1484.11.1-2004 的 B.1 標準中。此資料模型類型於 XSD 中被實作成 `real7Type` 公用類型，建立在 XML 架構的十進位原始資料類型基礎上。

```
<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 區間時間資料類型之實作

此類型的解釋定義於 IEEE 1484.11.1-2004 標準的 B.2 標準中。此資料模型類型於 XSD 中被實作成 `timeIntervalType` 公用類型，建立在 XML 架構原始資料類型持續時間的基礎上。此持續時間類型為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>
```

此類型在 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定義中的公用類型

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

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 文字字串類型

此類型是為字串而定義的，使得當中的空白不會由於一個 XML 實作而改變。literalStringType 類型用來當作其他三種類型的一個基礎抽象類型，分別為 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 課程狀態類型

此資料模型類型被定義為obsoleteStatusType公用類型，其定義為擁有一個標記值的列舉類型，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 標度分數類型

此資料模型類型被定義為建立在`real7Type`類型基礎上的數值類型，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 特定的文字字串類型

此類型並未明確地定義在 IEEE 1484.11.1-2004 標準中，卻被隱含在 ISO/ IEC 11404:1996 [B2] 中的字元字串資料類型中。因為 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、以字串為基礎的元件值不可以被改變。藉由 `literalStringType` 類型，它指定必須要保留空白。此類型被用來當作基礎的類型進一步定義 `localizedStringType` 類型，就如同下列三種類型：`literalString250Type`、`literalString1000Type`、和 `literalString4000Type`，此三種類型各自帶有 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應資料

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，作為學習者正確回應的複雜、資料模型類型之定義的一部份。他們按照英文字首排列。這些 XSD 的片段透過觀看 COCD XML 實例（參照附錄 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 對填寫之正確回應

在此群組定義下，每一個 fillMatches 元件皆為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:sequence is 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之正確回應

在此群組的定義中，likert的選擇是一個單獨、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之正確回應

在此群組定義中，每一個matchText元件皆為一個單獨、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 對配對之正確回應

在此群組定義中，每一個回應 `MatchingType` 類型的 `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`公用類型實作一個配對形態，此配對形態為配對組的集合。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公用類型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 對多重選擇之正確回應

此元件在資料模型中定義為短辨識符集合，每一個都代表一個選擇。外部集合為一 choice 元件的集合。xs:sequence 必須允許多於一個選項集合的實例。且不應解釋為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公用元件為bagOfChoiceTypes類型，為choice元件的集合。xs:unique必須限制集合中為單獨的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 類型為 choice 元件的集合，xs:sequence 不應解釋為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 對數值之正確回應

此群組定義包含兩個選擇性元件，用來指出正確回應之最小值和最大值。

```
<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 對其他之正確回應

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 對表現之正確回應

在此群組定義中，每一個 `correctPerformancePatternType` 類型的 `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 公用類型具體指明了單一次回應形態的編碼，該回應類型是一序列的步驟，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 對序列之正確回應

在此群組定義中，每一個 `stepSequenceType` 類型的 `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。且不應被解釋為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` 類型代表一個步驟序列。`xs:sequence` 在此應解釋為隱含特定的次序。

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 對真假之正確回應

在此群組定義中，真-假的選項為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公用類型實作互動類型的真假選項，該互動類型定義於IEEE 1484.11.1-2004標準的6.1.9.5 和 6.1.9.7。此公用類型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 對填入之學習者回應

在此群組定義中，學習者回應由零或多個 localizedString250Type 類型的 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 對likert之學習者回應

在此群組定義中，學習者回應為一單獨的、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 對long fill-in之學習者回應

在此群組的定義中，學習者回應為一單一的定位字串。

```
<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 對配對之學習者回應

在此群組的定義中，學習者回應為一配對的形態，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 對多重選擇之學習者回應

在此群組定義中，學習者回應為一識別符串列，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 對數值之學習者回應

在此群組定義中，學習者回應為一單一的數值。

```
<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 對其他之學習者回應

在此群組定義中，學習者回應為一單一文字字串。

```
<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 對表現之學習者回應

在此群組定義中，學習者回應為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 類型實作一個表現步驟的學習者回應之元件。

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 對序列之學習回應 (Learning response for sequencing)

在此群組定義中，學習者回應為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 對真假之學習者回應

在此群組定義中，學習者回應為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 配對回應類型

此類型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

(參考)

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.

英中名詞對照表

-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
collection	收集
content	教材
content object communication data (COCD)	教材物件溝通資料(COCD)
convention	規約
credit	學分

-D-

data model	資料模型
------------	------

-E-

element	元件
entry	進入
enumerated type	列舉類型
exit	退出
Extensible Markup Language(XML)	可延伸標示語言(XML)

-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
	response	回應
	root element	根元件
-S-		
	scaled passing score	標度及格分數
	schema	架構
	session time	交談時間
	smallest permitted maximum (SPM)	最小允許上限(SPM)
	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-		

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

Communication

Contents

1. Overview.....	145
2. Terms and definitions	145
3. Normative references.....	146
4. Conformance	146
5. XML binding	147
Annex A (informative) Bibliography.....	148
Annex B (normative) Normative XSD	149
Annex C (informative) An example COCD XML instance.....	197
Annex D (informative) Explanatory XSD notes	213
Annex E (informative) Internet availability of the XSD file and example instance....	257

1. Overview

1.1 Scope

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 Purpose

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. Terms and definitions

2.1 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.

- 2.1.1 content 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.

2.1.2 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.

2.1.3 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.

2.2 Acronyms 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. Normative references

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

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

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

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

4. Conformance

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.

A conforming COCD XML instance

- (1) Shall conform to the data-model requirements of IEEE Std 1484.11.1-2004.
- (2) Shall not contain any extensions to the data model defined in IEEE Std 1484.11.1-2004.
- (3) Shall be valid according to the XML Schema definition (XSD) specified in

Annex B.

- (4) Shall not contain any elements or attributes not defined in the XSD specified in Annex B.
- (5) 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.

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.
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 binding

The namespace for the XML binding is defined by the conforming XSD in Annex B and shall be

http://itsc.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.

Annex A (informative) Bibliography

(informative) Bibliography

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

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

Annex B (normative) Normative XSD

(normative)

Normative XSD

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

```
<?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:complexType>
</xs:element>
<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:all>
</xs:complexType>
</xs:element>
</xs:annotation>
</xs:documentation>
```

```

<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:element>

```

```

</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: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: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: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: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">

      Implements learner_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: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:complexType>
<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: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>
```

Figure B.1—Conforming XSD

Annex C (informative) An example COCD XML instance

(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.

```
<?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>

  <identifiant>urn:ostyn.com:LIQ2003090412345</identifiant>

  <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>
  </objective>
</objectives>

```

```
<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>
```

Figure C.1—An example COCD XML instance

Annex D (informative) Explanatory XSD notes

(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.

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>”.

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 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 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.

```
<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>
```

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

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

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.”

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:

- (1) Instantiate an XML document that contains an element for which the `collectionType` attribute is defined in the XSD in an XML Schema processor environment.
- (2) 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 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:

- (1) 6.1.9 Interactions: Each interaction requires an identifier, and the identifier has to be unique within the context of the content object.
- (2) 6.1.9.3 Objectives ID in an interaction record.
- (3) 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.
- (4) 6.1.18 Objectives: Each objective requires an identifier, and the identifier has to be unique within the context of the content object.

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 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 `maxOccurs="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 `spm` attribute.

An implementation might use the fixed `spm` attribute as follows:

- (1) Instantiate an XML document that contains an element for which the `spm` attribute is defined in the XSD in an XML Schema processor environment.
- (2) 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 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:annotation` element. 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.)

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

- (1) spm: This name denotes the SPM value for the element.
- (2) 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

- (1) Instantiate an XML fragment from the string contained in the appInfo element, if present, and extract the comment from that fragment.
- (2) Instantiate an XML fragment from the string contained in the comment.
- (3) Get the element spm or collectionType, if present, and inspect its value.

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.

```
<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>
```

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 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.

```
..
<xs:element name="commentsFromLearner">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="commentFromLearner"
      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 1484.11.12004.

```

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

```

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.

```

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

```

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.

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

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.

```
<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.3.6 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.

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

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.

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

```

    <xs:enumeration value="resume"/>

    <xs:enumeration value=""/>

  </xs:restriction>

</xs:simpleType>

</xs:element>

```

NOTE : An empty string ("") is represented in a COCD XML instance as an empty element (e.g.,

```
<cocd><entry/ ></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.

```

<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>

```

NOTE : An empty string ("") is represented in a COCD XML instance as an empty element (e.g.,

```
<cocd><exit/></ cocd>).
```

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.

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

```
<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:all>
```

```

</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.3.9.2 Encoding of 6.1.9.1 ID

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

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

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`.

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

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.

```
<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.3.9.5 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 `longIdentifierType`. A uniqueness constraint is applied.

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

```
</xs:unique>
</xs:element>
```

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

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

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`.

```
<xs:element name="timeStamp" type="dateTimeType" 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`.

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

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.

```
<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>
```

These groups are described in detail in D.7.

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`.

```
<xs:element name="weighting" type="real7Type" 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`.

```
<xs:element name="learnerResponse" type="learnerResponseType"
  minOccurs="0"/>
```

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.

```
<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>
```

These groups are described in detail in D.7.

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.

```
<xs:element name="result" type="interactionResultType"
  minOccurs="0"/>
```

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.

```
<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.3.9.14 Encoding of 6.1.9.9 Latency

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

```
<xs:element name="latency" type="timeIntervalType" 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`.

```
<xs:element name="description" type="localizedString250Type"
  minOccurs="0"/>
```

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.

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

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`.

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

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`.

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

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`.

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

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.

```
<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>

```

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.

```

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

```

D.3.13.2 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`.

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

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.

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

D.3.13.4 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.

```
<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.3.14 (Encoding of 6.1.14 Lesson status

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

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

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.

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

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`.

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

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.

```
<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.3.18 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 objective element. 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.

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

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

```
<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.3.18.1 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.

```
<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:all>
</xs:complexType>
```

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.

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

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`.

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

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`.

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

D.3.22 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.

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

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`.

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

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`.

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

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.

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

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.

```
<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.3.27 Encoding of 6.1.27 Total time

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

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

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 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.

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

```

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

```

D4.2 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.

```

<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.4.3 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.

```

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

```

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.

D4.4 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.

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

D4.5 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`.

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

D4.6 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 `base` attribute can be attached to it.

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

D4.7 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`.

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

D4.8 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.

```
<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>
```

D4.9 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 `spsattribute` can be attached to it.

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

D4.10 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.

```
<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>
```

D5 Implementation of other documented data types

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

D5.1 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`.

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

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.

D5.2 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.

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

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 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.

D6.1 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.

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

```
</xs:restriction>
```

```
</xs:simpleType>
```

D6.2 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.

```
<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>
```

D6.3 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.

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

```
  <xs:restriction base="real7Type">
```

```
    <xs:minInclusive value="-1"/>
```

```
    <xs:maxInclusive value="1"/>
```

```
  </xs:restriction>
```

```
</xs:simpleType>
```

D6.4 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.

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

D7 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).

D7.1 Correct responses for fill-in

In this group definition, each fillMatches element is a sequence of predefined matching strings. The use of xs:sequence is 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.

```
<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>

```

D7.2 Correct responses for likert

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

```

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

```

D7.3 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.

```
<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>
```

D7.4 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>
```

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>

```

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>

```

D7.5 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>

```

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="bagOfChoiceTypes">
  <xs:unique name="uniqueInChoicesIds">

```

```

    <xs:selector xpath="./choice"/>

    <xs:field xpath="."/>

  </xs:unique>

</xs:element>

```

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>

```

D7.6 Correct responses for numeric

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

```

<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>

```

D7.7 Correct response for other

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

```

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

```

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

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

D7.8 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>
```

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>

```

D7.9 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>

```

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>

```

D7.10 Correct response for true false

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

```

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

```

The global type `trueFalseType` implements true and false options for the interaction type `true_falsespecified` 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.

```

<xs:simpleType name="trueFalseType">
  <xs:restriction base="xs:token">

```

```

    <xs:enumeration value="true"/>

    <xs:enumeration value="false"/>

  </xs:restriction>

</xs:simpleType>

```

D7.11 Learner response for fill-in

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

```

<xs:group name="grpResponseFillIn">

  <xs:sequence>

    <xs:element name="fillString" type="localizedString250Type"

      minOccurs="0" maxOccurs="unbounded">

      </xs:element>

    </xs:sequence>

  </xs:group>

```

D7.12 Learner response for likert

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

```

<xs:group name="grpResponseLikert">

  <xs:sequence>

    <xs:element name="choice" type="shortIdentifierType"

      minOccurs="0"/>

    </xs:sequence>

  </xs:group>

```

D7.13 Learner response for long fill-in

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

```

<xs:group name="grpResponseLongFillIn">

  <xs:sequence>

```

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

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

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

D7.14 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.

```
<xs:group name="grpResponseMatching">
```

```
<xs:sequence>
```

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

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

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

D7.15 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.

```
<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>

```

D7.16 Learner response for numeric

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

```

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

```

D7.17 Learner response for other

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

```

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

```

D7.18 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>
```

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>
```

D7.19 Learning response for sequencing

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

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

D7.20 Learner response for true false

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

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

D7.21 Matching response type

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

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

Annex E (informative) Internet availability of the XSD file and
example instance

(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.

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，為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. 英中名詞對照表

-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
collection	收集
content	教材
content object communication data (COCD)	教材物件溝通資料(COCD)
convention	規約
credit	學分

-D-

data model	資料模型
------------	------

-E-

element	元件
entry	進入
enumerated type	列舉類型
exit	退出

	Extensible Markup Language(XML)	可延伸標示語言(XML)
-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
	response	回應
	root element	根元件
-S-	scaled passing score	標度及格分數
	schema	架構
	session time	交談時間
	smallest permitted maximum (SPM)	最小允許上限(SPM)
	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-		

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
收集	collection
教材	content
教材物件溝通資料(COCD)	content object communication data (COCD)
規約	convention
學分	credit
資料模型	data model
元件	element
進入	entry
列舉類型	enumerated type
退出	exit
可延伸標示語言(XML)	Extensible Markup Language(XML)
片段	fragment
識別符	identifier
實例	instance
互動	interaction
潛時	latency
文字	literal
最大允許時間	max time allowed
前綴	prefix
原始及格分數	raw passing score
回應	response
根元件	root element
標度及格分數	scaled passing score
架構	schema

交談時間	session time
最小允許上限(SPM)	smallest permitted maximum (SPM)
懸置資料	suspend data
時戳	time stamp
全球資訊網聯盟(W3C)	World Wide Web Consortium (W3C)
XML 架構定義(XSD)	XML Schema definition(XSD)

