

1 **IEEE 1484.11.3/Draft 6**
2 **Draft Standard for Learning Technology—**
3 **Extensible Markup Language (XML) Schema Binding**
4 **for Data Model for Content Object Communication**

5 Sponsor
6 Learning Technology Standards Committee
7 of the
8 IEEE Computer Society



9 **Abstract:** This Standard specifies a World Wide Web Consortium (W3C) Extensible
10 Markup Language (XML) binding of the data model defined in IEEE 1484.11.1–2004,
11 "IEEE Standard for Learning Technology – Data Model for Content Object Communica-
12 tion." The purpose of this Standard is to allow the creation of IEEE 1418.11.1–2004
13 data-model instances in XML. This Standard uses the W3C XML Schema definition lan-
14 guage as the encoding. This allows for interoperability and the exchange of data-model
15 instances between various systems.

16 **Keywords:** content object, Content Object Communication Data Model, IEEE
17 1484.11.1–2004, Extensible Markup Language, XML, XML binding, XML data instance,
18 XML Schema definition, W3C XML Schema definition language.

19 _____

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

22 Copyright © 2005 by the Institute of Electrical and Electronics Engineers, Inc.
23 All rights reserved. Published [date to be supplied](#). Printed in the United States of Amer-
24 ica.

25 This document is an unapproved draft of a proposed IEEE Standard. As such, this
26 document is subject to change. **USE AT YOUR OWN RISK!** Because this is an unap-
27 proved draft, this document must not be utilized for any conformance/compliance pur-
28 poses. Permission is hereby granted for IEEE Standards Committee participants to re-
29 produce this document for purposes of IEEE standardization activities only. Prior to
30 submitting this document to another standards development organization for standardi-
31 zation activities, permission must first be obtained from the Manager, Standards Licens-
32 ing and Contracts, IEEE Standards Activities Department. Other entities seeking per-
33 mission to reproduce this document, in whole or in part, must obtain permission from
34 the Manager, Standards Licensing and Contracts, IEEE Standards Activities Depart-
35 ment.

36 IEEE Standards Department
 37 Standards Licensing and Contracts
 38 445 Hoes Lane, P.O. Box 1331
 39 Piscataway, NJ 08855-1331, USA

40

41 *[Note: Information about IEEE LTSC P1484.12.3 can be found at:*

42

43 <http://ltsc.ieee.org/wg11/>

44

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

46 Introduction

47 (This introduction is not a part of P1484.11.3, Draft Standard for Learning Technology—
 48 Extensible Markup Language (XML) Schema Binding for Data Model for Content Object
 49 Communication.)

50 This Standard specifies a World Wide Web Consortium (W3C) Extensible Markup Language
 51 (XML) schema binding of the data model defined in IEEE 1484.11.1–2004, "IEEE Standard for
 52 Learning Technology – Data Model for Content Object Communication." The purpose of this
 53 Standard is to allow the creation of IEEE 1418.11.1–2004 data-model instances in XML. This
 54 Standard uses the W3C XML Schema definition language as the encoding. This allows for inter-
 55 operability and the exchange of data-model instances between various systems.

56 Participants

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

Kerry Blinco, *Chair*
 Tyde Richards, *Chair 2001 – 2004*
 Scott Lewis, *Technical Editor*

Jack Hyde

Rolf Lindner

Schawn Thropp

Tom King

Claude Ostyn

58 The following persons were on the balloting committee: (To be provided by IEEE editor at time
 59 of publication.)

60 Also included are the following nonvoting IEEE–SA Standards Board liaisons:

61 [To be supplied](#)

62 **Acknowledgements**

63 The working group would like to acknowledge Claude Ostyn for providing the original schema
64 and example instance that were used in the development of this Standard.

| | | |
|----|---|----|
| 65 | Contents | |
| 66 | 1. Overview..... | 1 |
| 67 | 1.1 Scope | 1 |
| 68 | 1.2 Purpose | 1 |
| 69 | 2. Normative references | 1 |
| 70 | 3. Definitions | 2 |
| 71 | 3.1 Acronyms and abbreviations | 2 |
| 72 | 4. Conformance..... | 2 |
| 73 | 5. XML binding | 3 |
| 74 | Annex A (informative) Bibliography | 4 |
| 75 | Annex B (normative) Normative XSD | 5 |
| 76 | Annex C (informative) An example COCD XML instance | 26 |
| 77 | Annex D (informative) Explanatory XSD notes..... | 34 |
| 78 | Annex E (informative) Internet availability of the XSD file and example instance..... | 63 |

79 **Draft Standard for Learning Technology—**
80 **Extensible Markup Language (XML) Schema Binding**
81 **for Data Model for Content Object Communication**

82 **1. Overview**

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

84 **1.1 Scope**

85 This Standard specifies a World Wide Web Consortium (W3C) Extensible Markup Language
86 (XML) Schema binding of the data model defined in IEEE 1484.11.1–2004, "IEEE Standard for
87 Learning Technology – Data Model for Content Object Communication."¹ An implementation
88 that conforms to this Standard shall conform to IEEE 1484.11.1–2004.

89 **1.2 Purpose**

90 The purpose of this Standard is to allow the creation of IEEE 1418.11.1–2004 data-model in-
91 stances in XML. This Standard uses the W3C XML Schema definition language to specify the
92 encoding of these data-model instances (see XML Schema Parts 1 and 2). This allows for inter-
93 operability and the exchange of data-model instances between various systems.

94 **2. Normative references**

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

98 IEEE 1484.11.1–2004, IEEE Standard for Learning Technology—Data Model for Content Ob-
99 ject Communication.

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

¹ For information on normative references, see Clause 2.

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

102 **3. Definitions**

103 For purposes of this Standard, the following terms and definitions apply. IEEE 100, *The Authori-*
104 *tative Dictionary of IEEE Standards Terms*, Seventh Edition [B1]², should be referenced for
105 terms not defined in this Clause.

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

110 **content object communication data Extensible Markup Language instance (COCD XML**
111 **instance)**: A particular XML representation of the data model defined in IEEE 1484.11.1–2004
112 that adheres to the requirements and constraints of an XML binding of the data model.

113 **Extensible Markup Language binding (XML binding)**: In this Standard, the method of encod-
114 ing the behaviors, attributes, and value spaces of data-model elements in W3C Extensible
115 Markup Language. This method is specified using the W3C XML Schema definition language.

116 **3.1 Acronyms and abbreviations**

117 COCD: content object communication data

118 SPM: smallest permitted maximum

119 W3C: World Wide Web Consortium

120 XML: Extensible Markup Language

121 XSD: XML Schema definition

122 **4. Conformance**

123 This Standard defines conforming IEEE 1481.11.1–2004 content object communication data
124 (COCD) instances in an XML binding. Hereafter, such instances are referred to as "COCD XML
125 instances."

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

² The numbers in brackets correspond to those of the bibliography in Annex A.

128 A conforming COCD XML instance

- 129 – Shall conform to the data-model requirements of IEEE 1484.11.1–2004;
- 130 – Shall not contain any extensions to the data model defined in IEEE 1484.11.1–2004;
- 131 – Shall be valid according to the XML Schema definition (XSD) specified in Annex B;
- 132 – Shall not contain any elements or attributes not defined in the XSD specified in Annex B;
- 133 and
- 134 – Shall consist of a single element and its descendants. The single element shall have the
- 135 name "cocd" as defined in the XSD specified in Annex B. The single element shall reside
- 136 within the scope of a namespace declaration using the namespace specified in Clause 5.

137 NOTES:

138 1—This Standard does not require that the COCD XML instance be an XML document. The instance
139 may be embedded at any depth in an XML data instance that includes elements from other namespaces.

140 2—IEEE 1484.11.1–2004 defines smallest permitted maximum (SPM) values. If a COCD XML instance
141 contains more than the SPM number of occurrences of a COCD element, implementers should be aware
142 that it is not guaranteed that an application will process more than the SPM number of occurrences of the
143 COCD element. If a COCD XML instance contains more than the SPM number of characters in a charac-
144 ter string, implementers should be aware that it is not guaranteed that an application will process more
145 than the SPM number of characters in the character string.

146 3—The W3C XML Schema definition language cannot express and enforce all the data-model require-
147 ments of IEEE 1484.11.1–2004 (e.g., the requirements for SPMs).

148 **5. XML binding**

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

150 `http://ltsc.ieee.org/xsd/1484_11_3`

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

152 An example COCD XML instance is given in Annex C.

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

156 **Annex A**

157 (informative)

158 **Bibliography**

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

160 [B2]ISO/IEC 11404:1996, Information technology – Programming languages, their environ-
161 ments and system software interfaces – Language-independent datatypes.

162 **Annex B**

163 (normative)

164 **Normative XSD**

165 Figure B1 shows the conforming XSD for the data model defined by IEEE 1484.11.1–2004.

```

166 <?xml version="1.0" encoding="UTF-8"?>
167 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
168   xmlns="http://ltsc.ieee.org/xsd/1484_11_3"
169   xmlns:t="http://ltsc.ieee.org/xsd/1484_11_3"
170   targetNamespace="http://ltsc.ieee.org/xsd/1484_11_3"
171   elementFormDefault="qualified" version="1484.11.3-1.0">
172   <xs:annotation>
173     <xs:documentation xml:lang="en">
174       This schema is specified in IEEE 1484.11.3-2005, "IEEE Standard
175       for Learning Technology - Extensible Markup Language (XML)
176       Binding for Data Model for Content Object Communication."
177       This schema is a World Wide Web Consortium (W3C) Extensible
178       Markup Language (XML) binding of the data model defined in IEEE
179       1484.11.1-2004, "IEEE Standard for Learning Technology - Data
180       Model for Content Object Communication."
181       The purpose of this schema is to allow the creation of IEEE
182       1418.11.1-2004 data-model instances in XML. This schema uses the
183       W3C XML Schema definition language as the encoding. This allows
184       for interoperability and the exchange of data-model instances
185       between various systems.
186       This schema shall not be modified but may be included in
187       derivative works.
188
189       Copyright (c) 2005 Institute of Electrical and Electronics
190       Engineers, Inc.
191
192       USE AT YOUR OWN RISK
193
194     </xs:documentation>
195   </xs:annotation>
196   <!-- -->
197   <!-- The first half of this document follows the order of the
198       Data Model document clauses. -->
199   <!-- -->
200   <xs:element name="cocd" type="cocdType"/>
201   <xs:complexType name="cocdType">
202     <xs:annotation>
203       <xs:documentation xml:lang="en">
204         Implements IEEE 1484.11.1-2004,
205         Clause 6.1: Content object communication
206       </xs:documentation>
207     </xs:annotation>
208     <xs:all>
209       <xs:element ref="commentsFromLearner" minOccurs="0"/>
210       <xs:element ref="commentsFromLMS" minOccurs="0"/>
211       <xs:element ref="completionStatus" minOccurs="0"/>
212       <xs:element ref="completionThreshold" minOccurs="0"/>
213       <xs:element ref="credit" minOccurs="0"/>

```

```

214 <xs:element ref="dataModelVersion" minOccurs="0"/>
215 <xs:element ref="entry" minOccurs="0"/>
216 <xs:element ref="exit" minOccurs="0"/>
217 <xs:element ref="interactions" minOccurs="0"/>
218 <xs:element ref="launchData" minOccurs="0"/>
219 <xs:element ref="learnerId" minOccurs="0"/>
220 <xs:element ref="learnerName" minOccurs="0"/>
221 <xs:element ref="learnerPreferenceData" minOccurs="0"/>
222 <xs:element ref="lessonStatus" minOccurs="0"/>
223 <xs:element ref="location" minOccurs="0"/>
224 <xs:element ref="maxTimeAllowed" minOccurs="0"/>
225 <xs:element ref="mode" minOccurs="0"/>
226 <xs:element ref="objectives" minOccurs="0"/>
227 <xs:element ref="progressMeasure" minOccurs="0"/>
228 <xs:element ref="rawPassingScore" minOccurs="0"/>
229 <xs:element ref="scaledPassingScore" minOccurs="0"/>
230 <xs:element ref="score" minOccurs="0"/>
231 <xs:element ref="sessionTime" minOccurs="0"/>
232 <xs:element ref="successStatus" minOccurs="0"/>
233 <xs:element ref="suspendData" minOccurs="0"/>
234 <xs:element ref="timeLimitAction" minOccurs="0"/>
235 <xs:element ref="totalTime" minOccurs="0"/>
236 </xs:all>
237 </xs:complexType>
238 <xs:element name="commentsFromLearner">
239 <xs:annotation>
240 <xs:documentation xml:lang="en">
241 Implements Clause 6.1.1: Comments from learner
242 </xs:documentation>
243 </xs:annotation>
244 <xs:complexType>
245 <xs:sequence>
246 <xs:element name="commentFromLearner" type="commentType"
247 minOccurs="0" maxOccurs="unbounded"/>
248 </xs:sequence>
249 <xs:attribute name="collectionType" fixed="array"/>
250 <xs:attribute name="spm" fixed="250"/>
251 </xs:complexType>
252 </xs:element>
253 <xs:element name="commentsFromLMS">
254 <xs:annotation>
255 <xs:documentation xml:lang="en">
256 Implements Clause 6.1.2: Comments from LMS
257 </xs:documentation>
258 </xs:annotation>
259 <xs:complexType>
260 <xs:sequence>
261 <xs:element name="commentFromLMS" type="commentType"
262 minOccurs="0" maxOccurs="unbounded"/>
263 </xs:sequence>
264 <xs:attribute name="collectionType" fixed="array"/>
265 <xs:attribute name="spm" fixed="100"/>
266 </xs:complexType>
267 </xs:element>
268 <xs:element name="completionStatus" type="completionStatusType">
269 <xs:annotation>
270 <xs:documentation xml:lang="en">
271 Implements Clause 6.1.3: Completion status
272 </xs:documentation>
273 </xs:annotation>

```

```

274 </xs:element>
275 <xs:element name="completionThreshold" type="progressMeasureType">
276   <xs:annotation>
277     <xs:documentation xml:lang="en">
278       Implements Clause 6.1.4: Completion threshold
279     </xs:documentation>
280   </xs:annotation>
281 </xs:element>
282 <xs:element name="credit">
283   <xs:annotation>
284     <xs:documentation xml:lang="en">
285       Implements Clause 6.1.5: Credit
286     </xs:documentation>
287   </xs:annotation>
288   <xs:simpleType>
289     <xs:restriction base="xs:token">
290       <xs:enumeration value="credit"/>
291       <xs:enumeration value="no_credit"/>
292     </xs:restriction>
293   </xs:simpleType>
294 </xs:element>
295 <xs:element name="dataModelVersion" type="literalString250Type">
296   <xs:annotation>
297     <xs:documentation xml:lang="en">
298       Implements Clause 6.1.6: Data model version
299     </xs:documentation>
300   </xs:annotation>
301 </xs:element>
302 <xs:element name="entry">
303   <xs:annotation>
304     <xs:documentation xml:lang="en">
305       Implements Clause 6.1.7: Entry
306     </xs:documentation>
307   </xs:annotation>
308   <xs:simpleType>
309     <xs:restriction base="xs:token">
310       <xs:enumeration value="ab_initio"/>
311       <xs:enumeration value="resume"/>
312       <xs:enumeration value=""/>
313     </xs:restriction>
314   </xs:simpleType>
315 </xs:element>
316 <xs:element name="exit">
317   <xs:annotation>
318     <xs:documentation xml:lang="en">
319       Implements Clause 6.1.8: Exit
320     </xs:documentation>
321   </xs:annotation>
322   <xs:simpleType>
323     <xs:restriction base="xs:token">
324       <xs:enumeration value="logout"/>
325       <xs:enumeration value="normal"/>
326       <xs:enumeration value="suspend"/>
327       <xs:enumeration value="timeout"/>
328       <xs:enumeration value=""/>
329     </xs:restriction>
330   </xs:simpleType>
331 </xs:element>
332 <xs:element name="interactions" type="interactionsType">
333   <xs:annotation>

```

```
334     <xs:documentation xml:lang="en">
335         Implements Clause 6.1.9: Interactions
336     </xs:documentation>
337 </xs:annotation>
338 </xs:element>
339 <xs:element name="launchData" type="literalString4000Type">
340     <xs:annotation>
341         <xs:documentation xml:lang="en">
342             Implements Clause 6.1.10: Launch data
343         </xs:documentation>
344     </xs:annotation>
345 </xs:element>
346 <xs:element name="learnerId" type="longIdentifierType">
347     <xs:annotation>
348         <xs:documentation xml:lang="en">
349             Implements Clause 6.1.11: Learner ID
350         </xs:documentation>
351     </xs:annotation>
352 </xs:element>
353 <xs:element name="learnerName" type="localizedString250Type">
354     <xs:annotation>
355         <xs:documentation xml:lang="en">
356             Implements Clause 6.1.12: Learner name
357         </xs:documentation>
358     </xs:annotation>
359 </xs:element>
360 <xs:element name="learnerPreferenceData"
361     type="learnerPreferenceType">
362     <xs:annotation>
363         <xs:documentation xml:lang="en">
364             Implements Clause 6.1.13: Learner preference data
365         </xs:documentation>
366     </xs:annotation>
367 </xs:element>
368 <xs:element name="lessonStatus" type="legacyStatusType">
369     <xs:annotation>
370         <xs:documentation xml:lang="en">
371             Implements Clause 6.1.14: Lesson status
372         </xs:documentation>
373     </xs:annotation>
374 </xs:element>
375 <xs:element name="location" type="literalString1000Type">
376     <xs:annotation>
377         <xs:documentation xml:lang="en">
378             Implements Clause 6.1.15: Location
379         </xs:documentation>
380     </xs:annotation>
381 </xs:element>
382 <xs:element name="maxTimeAllowed" type="timeIntervalType">
383     <xs:annotation>
384         <xs:documentation xml:lang="en">
385             Implements Clause 6.1.16: Max time allowed
386         </xs:documentation>
387     </xs:annotation>
388 </xs:element>
389 <xs:element name="mode">
390     <xs:annotation>
391         <xs:documentation xml:lang="en">
392             Implements Clause 6.1.17: Mode
393         </xs:documentation>
```

```

394     </xs:annotation>
395     <xs:simpleType>
396       <xs:restriction base="xs:token">
397         <xs:enumeration value="browse"/>
398         <xs:enumeration value="normal"/>
399         <xs:enumeration value="review"/>
400       </xs:restriction>
401     </xs:simpleType>
402 </xs:element>
403 <xs:element name="objectives" type="objectivesType">
404   <xs:annotation>
405     <xs:documentation xml:lang="en">
406       Implements Clause 6.1.18: Objectives
407     </xs:documentation>
408   </xs:annotation>
409   <xs:unique name="uniqueInSetOfObjectives">
410     <xs:selector xpath="./t:objective"/>
411     <xs:field xpath="t:identifier"/>
412   </xs:unique>
413 </xs:element>
414 <xs:element name="progressMeasure" type="progressMeasureType">
415   <xs:annotation>
416     <xs:documentation xml:lang="en">
417       Implements Clause 6.1.19: Progress measure
418     </xs:documentation>
419   </xs:annotation>
420 </xs:element>
421 <xs:element name="rawPassingScore" type="real7Type">
422   <xs:annotation>
423     <xs:documentation xml:lang="en">
424       Implements Clause 6.1.20: Raw passing score
425     </xs:documentation>
426   </xs:annotation>
427 </xs:element>
428 <xs:element name="scaledPassingScore" type="scaledScoreType">
429   <xs:annotation>
430     <xs:documentation xml:lang="en">
431       Implements Clause 6.1.21: Scaled passing score
432     </xs:documentation>
433   </xs:annotation>
434 </xs:element>
435 <xs:element name="score" type="scoreType">
436   <xs:annotation>
437     <xs:documentation xml:lang="en">
438       Implements Clause 6.1.22: Score
439     </xs:documentation>
440   </xs:annotation>
441 </xs:element>
442 <xs:element name="sessionTime" type="timeIntervalType">
443   <xs:annotation>
444     <xs:documentation xml:lang="en">
445       Implements Clause 6.1.23: Session time
446     </xs:documentation>
447   </xs:annotation>
448 </xs:element>
449 <xs:element name="successStatus" type="successStatusType">
450   <xs:annotation>
451     <xs:documentation xml:lang="en">
452       Implements Clause 6.1.24: Success status
453     </xs:documentation>

```

```

454     </xs:annotation>
455 </xs:element>
456 <xs:element name="suspendData" type="literalString4000Type">
457   <xs:annotation>
458     <xs:documentation xml:lang="en">
459       Implements Clause 6.1.25: Suspend data
460     </xs:documentation>
461   </xs:annotation>
462 </xs:element>
463 <xs:element name="timeLimitAction">
464   <xs:annotation>
465     <xs:documentation xml:lang="en">
466       Implements Clause 6.1.26: Time limit action
467     </xs:documentation>
468   </xs:annotation>
469   <xs:simpleType>
470     <xs:restriction base="xs:token">
471       <xs:enumeration value="continue_message"/>
472       <xs:enumeration value="continue_no_message"/>
473       <xs:enumeration value="exit_message"/>
474       <xs:enumeration value="exit_no_message"/>
475     </xs:restriction>
476   </xs:simpleType>
477 </xs:element>
478 <xs:element name="totalTime" type="timeIntervalType">
479   <xs:annotation>
480     <xs:documentation xml:lang="en">
481       Implements Clause 6.1.27: Total time
482     </xs:documentation>
483   </xs:annotation>
484 </xs:element>
485 <!-- -->
486 <!-- Global type declarations defined by numbered clauses in
487 IEEE 1484.11.1 -->
488 <!-- -->
489 <xs:complexType name="commentType">
490   <xs:annotation>
491     <xs:documentation xml:lang="en">
492       Implements Clause 6.2.1: Comment type
493     </xs:documentation>
494   </xs:annotation>
495   <xs:all>
496     <xs:element name="comment" type="localizedString4000Type"/>
497     <xs:element name="location" type="literalString1000Type"
498       minOccurs="0"/>
499     <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
500   </xs:all>
501 </xs:complexType>
502 <xs:simpleType name="completionStatusType">
503   <xs:annotation>
504     <xs:documentation xml:lang="en">
505       Implements Clause 6.2.2: Completion status type
506     </xs:documentation>
507   </xs:annotation>
508   <xs:restriction base="xs:token">
509     <xs:enumeration value="completed"/>
510     <xs:enumeration value="incomplete"/>
511     <xs:enumeration value="not_attempted"/>
512     <xs:enumeration value="unknown"/>
513   </xs:restriction>

```

```

514 </xs:simpleType>
515 <xs:simpleType name="dateTimeType">
516   <xs:annotation>
517     <xs:documentation xml:lang="en">
518       Implements Clause 6.2.3: Date time type
519     </xs:documentation>
520   </xs:annotation>
521   <xs:restriction base="xs:dateTime"/>
522 </xs:simpleType>
523 <xs:simpleType name="languageType">
524   <xs:annotation>
525     <xs:documentation xml:lang="en">
526       Implements Clause 6.2.4: Language type. Must remain simpleType
527       because it is used as value for attributes
528     </xs:documentation>
529   </xs:annotation>
530   <xs:restriction base="xs:language">
531     <xs:annotation>
532       <xs:appinfo>
533         <!-- <spm>250</spm> -->
534       </xs:appinfo>
535     </xs:annotation>
536   </xs:restriction>
537 </xs:simpleType>
538 <xs:complexType name="localizedStringType" abstract="true">
539   <xs:annotation>
540     <xs:documentation xml:lang="en">
541       Implements Clause 6.2.5: Localized string type.
542       The Localized String Type is implemented as several
543       variations with embedded SPM information.
544       The SPM is not enforced by XML validators but may
545       be useful for applications.
546     </xs:documentation>
547   </xs:annotation>
548   <xs:simpleContent>
549     <xs:extension base="literalStringType">
550       <xs:attribute name="lang" type="languageType"/>
551     </xs:extension>
552   </xs:simpleContent>
553 </xs:complexType>
554 <xs:complexType name="localizedString250Type">
555   <xs:annotation>
556     <xs:documentation xml:lang="en">
557       Implements Clause 6.2.5: Localized string type with SPM=250
558     </xs:documentation>
559   </xs:annotation>
560   <xs:simpleContent>
561     <xs:extension base="localizedStringType">
562       <xs:attribute name="spm" fixed="250"/>
563     </xs:extension>
564   </xs:simpleContent>
565 </xs:complexType>
566 <xs:complexType name="localizedString4000Type">
567   <xs:annotation>
568     <xs:documentation xml:lang="en">
569       Implements Clause 6.2.5: Localized string type with SPM=4000
570     </xs:documentation>
571   </xs:annotation>
572   <xs:simpleContent>
573     <xs:extension base="localizedStringType">

```

```

574     <xs:attribute name="spm" fixed="4000"/>
575   </xs:extension>
576 </xs:simpleContent>
577 </xs:complexType>
578 <xs:complexType name="longIdentifierType">
579   <xs:annotation>
580     <xs:documentation xml:lang="en">
581       Implements Clause 6.2.6: Long identifier type
582     </xs:documentation>
583   </xs:annotation>
584   <xs:simpleContent>
585     <xs:extension base="xs:anyURI">
586       <xs:attribute name="spm" fixed="4000"/>
587     </xs:extension>
588   </xs:simpleContent>
589 </xs:complexType>
590 <xs:simpleType name="progressMeasureType">
591   <xs:annotation>
592     <xs:documentation xml:lang="en">
593       Implements Clause 6.2.7: Progress measure type
594     </xs:documentation>
595   </xs:annotation>
596   <xs:restriction base="real7Type">
597     <xs:minInclusive value="0"/>
598     <xs:maxInclusive value="1"/>
599   </xs:restriction>
600 </xs:simpleType>
601 <xs:complexType name="scoreType">
602   <xs:annotation>
603     <xs:documentation xml:lang="en">
604       Implements Clause 6.2.8: Score type
605     </xs:documentation>
606   </xs:annotation>
607   <xs:all>
608     <xs:element name="scaled" type="scaledScoreType" minOccurs="0"/>
609     <xs:element name="max" type="real7Type" minOccurs="0"/>
610     <xs:element name="min" type="real7Type" minOccurs="0"/>
611     <xs:element name="raw" type="real7Type" minOccurs="0"/>
612   </xs:all>
613 </xs:complexType>
614 <xs:complexType name="shortIdentifierType">
615   <xs:annotation>
616     <xs:documentation xml:lang="en">
617       Implements Clause 6.2.9: Short identifier type
618     </xs:documentation>
619   </xs:annotation>
620   <xs:simpleContent>
621     <xs:extension base="xs:anyURI">
622       <xs:attribute name="spm" fixed="250"/>
623     </xs:extension>
624   </xs:simpleContent>
625 </xs:complexType>
626 <xs:simpleType name="successStatusType">
627   <xs:annotation>
628     <xs:documentation xml:lang="en">
629       Implements Clause 6.2.10: Success status type
630     </xs:documentation>
631   </xs:annotation>
632   <xs:restriction base="xs:token">
633     <xs:enumeration value="failed"/>

```



```

634     <xs:enumeration value="passed"/>
635     <xs:enumeration value="unknown"/>
636   </xs:restriction>
637 </xs:simpleType>
638 <xs:simpleType name="real7Type">
639   <xs:annotation>
640     <xs:documentation xml:lang="en">
641       As explained in IEEE 1484.11.1-2004, Annex B.1 Real data type
642     </xs:documentation>
643   </xs:annotation>
644   <xs:restriction base="xs:decimal"/>
645 </xs:simpleType>
646 <xs:simpleType name="timeIntervalType">
647   <xs:annotation>
648     <xs:documentation xml:lang="en">
649       As explained in IEEE 1484.11.1-2004, Annex B.2 Time interval
650       Data type
651     </xs:documentation>
652   </xs:annotation>
653   <xs:restriction base="xs:duration"/>
654 </xs:simpleType>
655 <!-- -->
656 <!-- Above this, things follow the order of the Data Model
657       document clauses. -->
658 <!-- ===== -->
659 <!-- Below this are things that did not fit neatly above. They are
660       organized as elements, attributes, groups, simple types, complex
661       types and alphabetically by name within each of those
662       categories. -->
663 <!-- -->
664 <!-- === ELEMENTS === -->
665 <!-- Organized in alphabetic order by element name -->
666 <!-- -->
667 <xs:element name="choices" type="setOfChoicesType">
668   <xs:annotation>
669     <xs:documentation xml:lang="en">
670       Set of short identifiers for interaction type "multiple
671       choice" as specified in 6.1.9.5: Correct response and 6.1.9.7:
672       Learner response.
673     </xs:documentation>
674   </xs:annotation>
675   <xs:unique name="uniqueInChoicesIds">
676     <xs:selector xpath="./t:choice"/>
677     <xs:field xpath="."/>
678   </xs:unique>
679 </xs:element>
680 <!-- -->
681 <!-- === ATTRIBUTES === -->
682 <!-- Organized in alphabetic order by attribute name -->
683 <!-- -->
684 <xs:attribute name="collectionType">
685   <xs:annotation>
686     <xs:documentation xml:lang="en">
687       The collectionType attribute is used to inject Data Model
688       information about aggregation that cannot be expressed in XML
689       schema. When defined for an element or type in this schema,
690       this attribute is given a fixed values. Even if the attribute
691       and value are not specified in an XML instance, the XML schema
692       processor makes them available to the processing application.
693     </xs:documentation>

```

```

694     </xs:annotation>
695     <xs:simpleType>
696         <xs:restriction base="xs:token">
697             <xs:enumeration value="bag"/>
698             <xs:enumeration value="array"/>
699             <xs:enumeration value="set"/>
700         </xs:restriction>
701     </xs:simpleType>
702 </xs:attribute>
703 <xs:attribute name="spm" type="xs:integer">
704     <xs:annotation>
705         <xs:documentation xml:lang="en">
706             The spm attribute is used to inject Data Model information
707             about SPM that cannot be expressed in XML schema. When defined
708             for an element or type in this schema, this attribute is given
709             a fixed values. Even if the attribute and value are not
710             specified in an XML instance, the XML schema processor makes
711             them available to the processing application.
712         </xs:documentation>
713     </xs:annotation>
714 </xs:attribute>
715 <!-- -->
716 <!-- === GROUPS === -->
717 <!-- Organized in alphabetic order by group name -->
718 <!-- -->
719 <xs:group name="grpCorrectFillIn">
720     <xs:sequence>
721         <xs:annotation>
722             <xs:appinfo>
723                 <!--
724                 <spm>5</spm><collectionType>bag</collectionType>
725                 -->
726             </xs:appinfo>
727         </xs:annotation>
728         <xs:element name="fillMatches" minOccurs="0"
729             maxOccurs="unbounded">
730             <xs:complexType>
731                 <xs:sequence>
732                     <xs:element name="matchText" type="localizedString250Type"
733                         maxOccurs="unbounded"/>
734                 </xs:sequence>
735                 <xs:attribute name="caseMatters" type="trueFalseType"
736                     use="optional" default="false"/>
737                 <xs:attribute name="orderMatters" type="trueFalseType"
738                     use="optional" default="true"/>
739                 <xs:attribute name="collectionType" fixed="array"/>
740                 <xs:attribute name="spm" fixed="10"/>
741             </xs:complexType>
742         </xs:element>
743     </xs:sequence>
744 </xs:group>
745 <xs:group name="grpCorrectLikert">
746     <xs:sequence>
747         <xs:element name="choice" type="shortIdentifierType"
748             minOccurs="0"/>
749     </xs:sequence>
750 </xs:group>
751 <xs:group name="grpCorrectLongFillIn">
752     <xs:sequence>
753         <xs:annotation>

```

```

754     <xs:appinfo>
755         <!--
756             <spm>5</spm><collectionType>bag</collectionType>
757         -->
758     </xs:appinfo>
759 </xs:annotation>
760 <xs:element name="matchText" maxOccurs="unbounded">
761     <xs:complexType>
762         <xs:simpleContent>
763             <xs:extension base="localizedString4000Type">
764                 <xs:attribute name="caseMatters" type="trueFalseType"
765                     use="optional" default="false"/>
766             </xs:extension>
767         </xs:simpleContent>
768     </xs:complexType>
769 </xs:element>
770 </xs:sequence>
771 </xs:group>
772 <xs:group name="grpCorrectMatching">
773     <xs:sequence>
774         <xs:annotation>
775             <xs:appinfo>
776                 <!--
777                     <spm>5</spm><collectionType>bag</collectionType>
778                 -->
779             </xs:appinfo>
780         </xs:annotation>
781         <xs:element name="matchPattern" type="matchingPairsType"
782             maxOccurs="unbounded" />
783     </xs:sequence>
784 </xs:group>
785 <xs:group name="grpCorrectMultipleChoice">
786     <xs:sequence>
787         <xs:annotation>
788             <xs:appinfo>
789                 <!--
790                     <spm>10</spm><collectionType>set</collectionType>
791                 -->
792             </xs:appinfo>
793         </xs:annotation>
794         <xs:element ref="choices" minOccurs="0" maxOccurs="unbounded" />
795     </xs:sequence>
796 </xs:group>
797 <xs:group name="grpCorrectNumeric">
798     <xs:sequence>
799         <xs:element name="min" type="real7Type" minOccurs="0" />
800         <xs:element name="max" type="real7Type" minOccurs="0" />
801     </xs:sequence>
802 </xs:group>
803 <xs:group name="grpCorrectOther">
804     <xs:sequence>
805         <xs:element name="correctOther" type="literalString4000Type" />
806     </xs:sequence>
807 </xs:group>
808 <xs:group name="grpCorrectPerformance">
809     <xs:sequence>
810         <xs:annotation>
811             <xs:appinfo>
812                 <!--
813                     <spm>5</spm><collectionType>bag</collectionType>

```

```

814     -->
815     </xs:appinfo>
816     </xs:annotation>
817     <xs:element name="performancePattern"
818       type="correctPerformancePatternType" maxOccurs="unbounded"/>
819   </xs:sequence>
820 </xs:group>
821 <xs:group name="grpCorrectSequencing">
822   <xs:sequence>
823     <xs:annotation>
824       <xs:appinfo>
825         <!--
826           <spm>5</spm><collectionType>bag</collectionType>
827         -->
828       </xs:appinfo>
829     </xs:annotation>
830     <xs:element name="stepSequence" type="stepSequenceType"
831       maxOccurs="unbounded"/>
832   </xs:sequence>
833 </xs:group>
834 <xs:group name="grpCorrectTrueFalse">
835   <xs:sequence>
836     <xs:element name="trueOrFalse" type="trueFalseType"/>
837   </xs:sequence>
838 </xs:group>
839 <!-- variant groups for interaction responses -->
840 <xs:group name="grpResponseFillIn">
841   <xs:sequence>
842     <xs:annotation>
843       <xs:appinfo>
844         <!--
845           <spm>10</spm><collectionType>array</collectionType>
846         -->
847       </xs:appinfo>
848     </xs:annotation>
849     <xs:element name="fillString" type="localizedString250Type"
850       minOccurs="0" maxOccurs="unbounded">
851       <xs:annotation>
852         <xs:appinfo>
853           <!-- <spm>250</spm> -->
854         </xs:appinfo>
855       </xs:annotation>
856     </xs:element>
857   </xs:sequence>
858 </xs:group>
859 <xs:group name="grpResponseLikert">
860   <xs:sequence>
861     <xs:element name="choice" type="shortIdentifierType"
862       minOccurs="0"/>
863   </xs:sequence>
864 </xs:group>
865 <xs:group name="grpResponseLongFillIn">
866   <xs:sequence>
867     <xs:element name="longFillString" type="localizedString4000Type"
868       minOccurs="0"/>
869   </xs:sequence>
870 </xs:group>
871 <xs:group name="grpResponseMatching">
872   <xs:sequence>
873     <xs:element name="matchPattern" type="matchingPairsType"/>

```

```

874     </xs:sequence>
875 </xs:group>
876 <xs:group name="grpResponseMultipleChoice">
877     <xs:sequence>
878         <xs:element ref="choices"/>
879     </xs:sequence>
880 </xs:group>
881 <xs:group name="grpResponseNumeric">
882     <xs:sequence>
883         <xs:element name="number" type="real7Type" minOccurs="0"/>
884     </xs:sequence>
885 </xs:group>
886 <xs:group name="grpResponseOther">
887     <xs:sequence>
888         <xs:element name="responseOther" type="literalString4000Type"/>
889     </xs:sequence>
890 </xs:group>
891 <!-- -->
892 <xs:group name="grpResponsePerformance">
893     <xs:annotation>
894         <xs:documentation xml:lang="en">
895             The learner response for interaction type "performance"
896             as specified in 6.1.9.7: Learner response
897         </xs:documentation>
898     </xs:annotation>
899     <xs:sequence>
900         <xs:annotation>
901             <xs:appinfo>
902                 <!--
903                 <spm>250</spm><collectionType>array</collectionType>
904                 -->
905             </xs:appinfo>
906         </xs:annotation>
907         <xs:element name="step" type="learnerPerformanceStepType"
908             minOccurs="0" maxOccurs="unbounded"/>
909     </xs:sequence>
910 </xs:group>
911 <xs:group name="grpResponseSequencing">
912     <xs:sequence>
913         <xs:element name="steps" type="stepSequenceType" minOccurs="0"/>
914     </xs:sequence>
915 </xs:group>
916 <xs:group name="grpResponseTrueFalse">
917     <xs:sequence>
918         <xs:element name="trueOrFalse" type="trueFalseType"
919             minOccurs="0"/>
920     </xs:sequence>
921 </xs:group>
922 <!-- -->
923 <!-- === SIMPLE TYPES === -->
924 <!-- Organized in alphabetic order by type name -->
925 <!-- -->
926 <xs:simpleType name="interactionResultType">
927     <xs:annotation>
928         <xs:documentation xml:lang="en">
929             Reusable type definition used for 6.1.9.8: Result
930             The value of result can be either a numeric value
931             or a specified token. This element uses xs:union to avoid
932             having to define sub-elements with arbitrary names.
933         </xs:documentation>

```

```

934     </xs:annotation>
935     <xs:union memberTypes="real7Type interactionResultTokenType"/>
936 </xs:simpleType>
937 <xs:simpleType name="interactionResultTokenType">
938     <xs:restriction base="xs:token">
939         <xs:enumeration value="correct"/>
940         <xs:enumeration value="incorrect"/>
941         <xs:enumeration value="neutral"/>
942         <xs:enumeration value="unanticipated"/>
943     </xs:restriction>
944 </xs:simpleType>
945 <xs:simpleType name="interactionTypeType">
946     <xs:annotation>
947         <xs:documentation xml:lang="en">
948             Implements Clause 6.1.9.2: Type
949         </xs:documentation>
950     </xs:annotation>
951     <xs:restriction base="xs:token">
952         <xs:enumeration value="true_false"/>
953         <xs:enumeration value="multiple_choice"/>
954         <xs:enumeration value="fill_in"/>
955         <xs:enumeration value="long_fill_in"/>
956         <xs:enumeration value="likert"/>
957         <xs:enumeration value="matching"/>
958         <xs:enumeration value="performance"/>
959         <xs:enumeration value="sequencing"/>
960         <xs:enumeration value="numeric"/>
961         <xs:enumeration value="other"/>
962     </xs:restriction>
963 </xs:simpleType>
964 <xs:simpleType name="literalStringType">
965     <xs:annotation>
966         <xs:documentation xml:lang="en">
967             Ensures preservation of whitespace
968         </xs:documentation>
969     </xs:annotation>
970     <xs:restriction base="xs:string">
971         <xs:whiteSpace value="preserve"/>
972     </xs:restriction>
973 </xs:simpleType>
974 <xs:simpleType name="legacyStatusType">
975     <xs:restriction base="xs:token">
976         <xs:enumeration value="browsed"/>
977         <xs:enumeration value="completed"/>
978         <xs:enumeration value="failed"/>
979         <xs:enumeration value="incomplete"/>
980         <xs:enumeration value="not_attempted"/>
981         <xs:enumeration value="passed"/>
982     </xs:restriction>
983 </xs:simpleType>
984 <xs:simpleType name="scaledScoreType">
985     <xs:restriction base="real7Type">
986         <xs:minInclusive value="-1"/>
987         <xs:maxInclusive value="1"/>
988     </xs:restriction>
989 </xs:simpleType>
990 <xs:simpleType name="trueFalseType">
991     <xs:annotation>
992         <xs:documentation xml:lang="en">
993             True and false options for interaction type "true_false"

```

```

994         as specified in 6.1.9.5: Correct response and 6.1.9.7:
995         Learner response.
996         Also used for tokens for various other boolean elements.
997     </xs:documentation>
998 </xs:annotation>
999     <xs:restriction base="xs:token">
1000         <xs:enumeration value="true"/>
1001         <xs:enumeration value="false"/>
1002     </xs:restriction>
1003 </xs:simpleType>
1004 <!-- -->
1005 <!-- === COMPLEX TYPES === -->
1006 <!-- Organized in alphabetic order by type name -->
1007 <!-- -->
1008 <xs:complexType name="setOfChoicesType">
1009     <xs:annotation>
1010         <xs:documentation xml:lang="en">
1011             Set of short identifiers for interaction type "multiple choice"
1012             as specified in 6.1.9.5: Correct response and 6.1.9.7: Learner
1013             response.
1014         </xs:documentation>
1015     </xs:annotation>
1016     <xs:sequence>
1017         <xs:element name="choice" type="shortIdentifierType"
1018             minOccurs="0" maxOccurs="unbounded"/>
1019     </xs:sequence>
1020     <xs:attribute name="collectionType" fixed="set"/>
1021     <xs:attribute name="spm" fixed="36"/>
1022 </xs:complexType>
1023 <xs:complexType name="correctPerformancePatternType">
1024     <xs:annotation>
1025         <xs:documentation xml:lang="en">
1026             A correct response record (order_matters + answers)
1027             For interaction type "performance"
1028             as specified in 6.1.9.5: Correct response
1029         </xs:documentation>
1030     </xs:annotation>
1031     <xs:sequence>
1032         <xs:element name="step" minOccurs="0" maxOccurs="unbounded">
1033             <xs:complexType>
1034                 <xs:all>
1035                     <xs:element name="stepName" type="shortIdentifierType"
1036                         minOccurs="0"/>
1037                     <xs:element name="stepAnswer" minOccurs="0">
1038                         <xs:complexType>
1039                             <xs:choice>
1040                                 <xs:element name="literal"
1041                                     type="literalString250Type" minOccurs="0"/>
1042                                 <xs:element name="numeric" minOccurs="0">
1043                                     <xs:complexType>
1044                                         <xs:attribute name="min" type="real7Type"/>
1045                                         <xs:attribute name="max" type="real7Type"/>
1046                                     </xs:complexType>
1047                                 </xs:element>
1048                             </xs:choice>
1049                         </xs:complexType>
1050                     </xs:element>
1051                 </xs:all>
1052             </xs:complexType>
1053         </xs:element>

```

```

1054 </xs:sequence>
1055 <xs:attribute name="orderMatters" type="trueFalseType"
1056   use="optional" default="true"/>
1057 <xs:attribute name="collectionType" fixed="array"/>
1058 <xs:attribute name="spm" fixed="250"/>
1059 </xs:complexType>
1060 <xs:complexType name="correctResponsesType">
1061   <xs:annotation>
1062     <xs:documentation xml:lang="en">
1063       Implements Clause 6.1.9.5: Correct responses.
1064       Note: It is up to the implementation to choose the correct
1065       group to match the interaction type. This correspondence cannot
1066       be expressed or validated using XML Schema.
1067     </xs:documentation>
1068   </xs:annotation>
1069   <xs:choice>
1070     <xs:group ref="grpCorrectTrueFalse"/>
1071     <xs:group ref="grpCorrectMultipleChoice"/>
1072     <xs:group ref="grpCorrectFillIn"/>
1073     <xs:group ref="grpCorrectLongFillIn"/>
1074     <xs:group ref="grpCorrectLikert"/>
1075     <xs:group ref="grpCorrectMatching"/>
1076     <xs:group ref="grpCorrectPerformance"/>
1077     <xs:group ref="grpCorrectSequencing"/>
1078     <xs:group ref="grpCorrectNumeric"/>
1079     <xs:group ref="grpCorrectOther"/>
1080   </xs:choice>
1081 </xs:complexType>
1082 <xs:complexType name="interactionType">
1083   <xs:annotation>
1084     <xs:documentation xml:lang="en">
1085       Reusable type to implements a single interaction record as
1086       defined in Clause 6.1.9.
1087     </xs:documentation>
1088   </xs:annotation>
1089   <xs:all>
1090     <xs:element name="identifier" type="longIdentifierType">
1091       <xs:annotation>
1092         <xs:documentation xml:lang="en">
1093           Implements Clause 6.1.9.1: ID
1094         </xs:documentation>
1095       </xs:annotation>
1096     </xs:element>
1097     <xs:element name="type" type="interactionTypeType">
1098       <xs:annotation>
1099         <xs:documentation xml:lang="en">
1100           Implements Clause 6.1.9.2: Type
1101         </xs:documentation>
1102       </xs:annotation>
1103     </xs:element>
1104     <xs:element name="objectiveIds" type="objectiveIdsType"
1105       minOccurs="0">
1106       <xs:annotation>
1107         <xs:documentation xml:lang="en">
1108           Implements Clause 6.1.9.3: Objectives ID
1109         </xs:documentation>
1110       </xs:annotation>
1111       <xs:unique name="uniqueInObjectivesIds">
1112         <xs:selector xpath="./t:objectiveId"/>
1113         <xs:field xpath="."/>

```



```

1114     </xs:unique>
1115   </xs:element>
1116   <xs:element name="timeStamp" type="dateTimeType" minOccurs="0">
1117     <xs:annotation>
1118       <xs:documentation xml:lang="en">
1119         Implements Clause 6.1.9.4: Time stamp
1120       </xs:documentation>
1121     </xs:annotation>
1122   </xs:element>
1123   <xs:element name="correctResponses" type="correctResponsesType"
1124     minOccurs="0">
1125     <xs:annotation>
1126       <xs:documentation xml:lang="en">
1127         Implements Clause 6.1.9.5: Correct responses
1128       </xs:documentation>
1129     </xs:annotation>
1130   </xs:element>
1131   <xs:element name="weighting" type="real7Type" minOccurs="0">
1132     <xs:annotation>
1133       <xs:documentation xml:lang="en">
1134         Implements Clause 6.1.9.6: Weighting
1135       </xs:documentation>
1136     </xs:annotation>
1137   </xs:element>
1138   <xs:element name="learnerResponse" type="learnerResponseType"
1139     minOccurs="0">
1140     <xs:annotation>
1141       <xs:documentation xml:lang="en">
1142         Implements Clause 6.1.9.7: Learner response
1143       </xs:documentation>
1144     </xs:annotation>
1145   </xs:element>
1146   <xs:element name="result" type="interactionResultType"
1147     minOccurs="0">
1148     <xs:annotation>
1149       <xs:documentation xml:lang="en">
1150         Implements Clause 6.1.9.8: Result
1151       </xs:documentation>
1152     </xs:annotation>
1153   </xs:element>
1154   <xs:element name="latency" type="timeIntervalType" minOccurs="0">
1155     <xs:annotation>
1156       <xs:documentation xml:lang="en">
1157         Implements Clause 6.1.9.9: Latency
1158       </xs:documentation>
1159     </xs:annotation>
1160   </xs:element>
1161   <xs:element name="description" type="localizedString250Type"
1162     minOccurs="0"/>
1163 </xs:all>
1164 </xs:complexType>
1165 <xs:complexType name="interactionsType">
1166   <xs:sequence>
1167     <xs:element name="interaction" type="interactionType"
1168       minOccurs="0" maxOccurs="unbounded"/>
1169   </xs:sequence>
1170   <xs:attribute name="collectionType" fixed="bag"/>
1171   <xs:attribute name="spm" fixed="250"/>
1172 </xs:complexType>
1173 <xs:complexType name="learnerPerformanceStepType">

```

```

1174 <xs:all>
1175   <xs:element name="stepName" type="shortIdentifierType"
1176     minOccurs="0"/>
1177   <xs:element name="stepAnswer" minOccurs="0">
1178     <xs:complexType>
1179       <xs:choice>
1180         <xs:element name="literal" type="literalString250Type"
1181           minOccurs="0"/>
1182         <xs:element name="numeric" type="real7Type" minOccurs="0"/>
1183       </xs:choice>
1184     </xs:complexType>
1185   </xs:element>
1186 </xs:all>
1187 </xs:complexType>
1188 <xs:complexType name="learnerPreferenceType">
1189   <xs:annotation>
1190     <xs:documentation xml:lang="en">
1191       Implements learner_preference_type in Clause 6.1.13:
1192       Learner preference data
1193     </xs:documentation>
1194   </xs:annotation>
1195   <xs:all>
1196     <xs:element name="audioLevel" minOccurs="0">
1197       <xs:annotation>
1198         <xs:documentation xml:lang="en">
1199           Implements Clause 6.1.13.1: Audio level
1200         </xs:documentation>
1201       </xs:annotation>
1202       <xs:simpleType>
1203         <xs:restriction base="real7Type">
1204           <xs:minInclusive value="0"/>
1205         </xs:restriction>
1206       </xs:simpleType>
1207     </xs:element>
1208     <xs:element name="language" type="languageType" minOccurs="0">
1209       <xs:annotation>
1210         <xs:documentation xml:lang="en">
1211           Implements Clause 6.1.13.2: Language
1212         </xs:documentation>
1213       </xs:annotation>
1214     </xs:element>
1215     <xs:element name="deliverySpeed" minOccurs="0">
1216       <xs:annotation>
1217         <xs:documentation xml:lang="en">
1218           Implements Clause 6.1.13.3: Delivery speed
1219         </xs:documentation>
1220       </xs:annotation>
1221       <xs:simpleType>
1222         <xs:restriction base="real7Type">
1223           <xs:minInclusive value="0"/>
1224         </xs:restriction>
1225       </xs:simpleType>
1226     </xs:element>
1227     <xs:element name="audioCaptioning" minOccurs="0">
1228       <xs:annotation>
1229         <xs:documentation xml:lang="en">
1230           Implements Clause 6.1.13.4: Audio captioning
1231         </xs:documentation>
1232       </xs:annotation>
1233     </xs:element>

```

```

1234     <xs:restriction base="xs:token">
1235         <xs:enumeration value="off"/>
1236         <xs:enumeration value="no_change"/>
1237         <xs:enumeration value="on"/>
1238     </xs:restriction>
1239 </xs:simpleType>
1240 </xs:element>
1241 </xs:all>
1242 </xs:complexType>
1243 <xs:complexType name="learnerResponseType">
1244     <xs:annotation>
1245         <xs:documentation xml:lang="en">
1246             Implements Clause 6.1.9.7: Learner response.
1247             Note: It is up to the implementation to choose the correct
1248             group or element name to match the interaction type. This
1249             correspondence cannot be expressed or validated using XML
1250             Schema.
1251         </xs:documentation>
1252     </xs:annotation>
1253     <xs:choice>
1254         <xs:group ref="grpResponseTrueFalse"/>
1255         <xs:group ref="grpResponseMultipleChoice"/>
1256         <xs:group ref="grpResponseFillIn"/>
1257         <xs:group ref="grpResponseLongFillIn"/>
1258         <xs:group ref="grpResponseLikert"/>
1259         <xs:group ref="grpResponseMatching"/>
1260         <xs:group ref="grpResponsePerformance"/>
1261         <xs:group ref="grpResponseSequencing"/>
1262         <xs:group ref="grpResponseNumeric"/>
1263         <xs:group ref="grpResponseOther"/>
1264     </xs:choice>
1265 </xs:complexType>
1266 <xs:complexType name="literalString250Type">
1267     <xs:annotation>
1268         <xs:documentation xml:lang="en">
1269             Implement any literal string with SPM=250
1270         </xs:documentation>
1271     </xs:annotation>
1272     <xs:simpleContent>
1273         <xs:extension base="literalStringType">
1274             <xs:attribute name="spm" fixed="250"/>
1275         </xs:extension>
1276     </xs:simpleContent>
1277 </xs:complexType>
1278 <xs:complexType name="literalString1000Type">
1279     <xs:annotation>
1280         <xs:documentation xml:lang="en">
1281             Implement any literal string with SPM=1000
1282         </xs:documentation>
1283     </xs:annotation>
1284     <xs:simpleContent>
1285         <xs:extension base="literalStringType">
1286             <xs:attribute name="spm" fixed="1000"/>
1287         </xs:extension>
1288     </xs:simpleContent>
1289 </xs:complexType>
1290 <xs:complexType name="literalString4000Type">
1291     <xs:annotation>
1292         <xs:documentation xml:lang="en">
1293             Implement any literal string with SPM=4000

```

```

1294     </xs:documentation>
1295 </xs:annotation>
1296 <xs:simpleContent>
1297   <xs:extension base="literalStringType">
1298     <xs:attribute name="spm" fixed="4000"/>
1299   </xs:extension>
1300 </xs:simpleContent>
1301 </xs:complexType>
1302 <xs:complexType name="matchingPairType">
1303   <xs:annotation>
1304     <xs:documentation xml:lang="en">
1305       A pair of matched short identifiers for interaction type
1306       "matching" as specified in 6.1.9.5: Correct response and
1307       6.1.9.7: Learner response.
1308     </xs:documentation>
1309   </xs:annotation>
1310   <xs:all>
1311     <xs:element name="source" type="shortIdentifierType"/>
1312     <xs:element name="target" type="shortIdentifierType"/>
1313   </xs:all>
1314 </xs:complexType>
1315 <xs:complexType name="matchingPairsType">
1316   <xs:sequence>
1317     <xs:annotation>
1318       <xs:documentation xml:lang="en">
1319         A collection of 0 or more instances of matchingPairType.
1320       </xs:documentation>
1321     </xs:annotation>
1322     <xs:element name="pair" type="matchingPairType" minOccurs="0"
1323       maxOccurs="unbounded"/>
1324   </xs:sequence>
1325   <xs:attribute name="collectionType" fixed="bag"/>
1326   <xs:attribute name="spm" fixed="36"/>
1327 </xs:complexType>
1328 <xs:complexType name="objectiveIdsType">
1329   <xs:sequence>
1330     <xs:element name="objectiveId" type="longIdentifierType"
1331       minOccurs="0" maxOccurs="unbounded"/>
1332   </xs:sequence>
1333   <xs:attribute name="collectionType" fixed="array"/>
1334   <xs:attribute name="spm" fixed="10"/>
1335 </xs:complexType>
1336 <xs:complexType name="objectivesType">
1337   <xs:sequence>
1338     <xs:element name="objective" type="objectiveType" minOccurs="0"
1339       maxOccurs="unbounded"/>
1340   </xs:sequence>
1341   <xs:attribute name="collectionType" fixed="set"/>
1342   <xs:attribute name="spm" fixed="100"/>
1343 </xs:complexType>
1344 <xs:complexType name="objectiveType">
1345   <xs:annotation>
1346     <xs:documentation xml:lang="en">
1347       Reusable type to implement the objective_type record
1348       defined in Clause 6.1.18
1349     </xs:documentation>
1350   </xs:annotation>
1351   <xs:all>
1352     <xs:element name="identifier" type="longIdentifierType"/>
1353     <xs:element name="score" type="scoreType" minOccurs="0"/>

```

```

1354     <xs:element name="status" type="legacyStatusType" minOccurs="0"/>
1355     <xs:element name="progressMeasure" type="progressMeasureType"
1356       minOccurs="0"/>
1357     <xs:element name="completionStatus" type="completionStatusType"
1358       minOccurs="0"/>
1359     <xs:element name="successStatus" type="successStatusType"
1360       minOccurs="0"/>
1361     <xs:element name="description" type="localizedString250Type"
1362       minOccurs="0"/>
1363   </xs:all>
1364 </xs:complexType>
1365 <xs:complexType name="responseOtherType">
1366   <xs:simpleContent>
1367     <xs:extension base="literalString4000Type"/>
1368   </xs:simpleContent>
1369 </xs:complexType>
1370 <xs:complexType name="stepSequenceType">
1371   <xs:annotation>
1372     <xs:documentation xml:lang="en">
1373       A sequence of steps for interaction type "sequencing"
1374       as specified in 6.1.9.5: Correct response and 6.1.9.7:
1375       Learner response.
1376       The same identifier may appear more than once in the list,
1377       because a step may be repeated in an interaction.
1378     </xs:documentation>
1379   </xs:annotation>
1380   <xs:sequence>
1381     <xs:element name="step" type="shortIdentifierType" minOccurs="0"
1382       maxOccurs="unbounded"/>
1383   </xs:sequence>
1384   <xs:attribute name="collectionType" fixed="array"/>
1385   <xs:attribute name="spm" fixed="36"/>
1386 </xs:complexType>
1387 </xs:schema>

```

1388

Figure B1—Conforming XSD

1389 **Annex C**

1390 (informative)

1391 **An example COCD XML instance**

1392 Figure C1 shows a COCD XML instance that instantiates all of the elements and attributes de-
 1393 fined in the XSD in Annex B.

```

1394 <?xml version="1.0" encoding="UTF-8"?>
1395 <cocd xmlns="http://ltsc.ieee.org/xsd/1484_11_3"
1396     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1397     xsi:schemaLocation="http://ltsc.ieee.org/xsd/1484_11_3
1398     ieee_1484_11_3_2005.xsd">
1399   <commentsFromLearner>
1400     <commentFromLearner>
1401       <comment lang="en-us">Having a good time</comment>
1402       <location>Somewhere</location>
1403       <timeStamp>2005-10-17T09:30:47-05:00</timeStamp>
1404     </commentFromLearner>
1405   </commentsFromLearner>
1406   <commentsFromLMS>
1407     <commentFromLMS>
1408       <comment lang="en-us">Have a good time</comment>
1409       <location>Wherever it may be</location>
1410       <timeStamp>2005-10-17T08:30:02-05:00</timeStamp>
1411     </commentFromLMS>
1412   </commentsFromLMS>
1413   <completionStatus>completed</completionStatus>
1414   <completionThreshold>0.9</completionThreshold>
1415   <credit>credit</credit>
1416   <dataModelVersion>1484.11.1</dataModelVersion>
1417   <entry>ab_initio</entry>
1418   <exit>logout</exit>
1419   <launchData>Nothing special</launchData>
1420   <learnerId>1234-foobar-1234</learnerId>
1421   <learnerName lang="en-us">John Doe</learnerName>
1422   <learnerPreferenceData>
1423     <audioLevel>1</audioLevel>
1424     <language>en-us</language>
1425     <deliverySpeed>1.25</deliverySpeed>
1426     <audioCaptioning>off</audioCaptioning>
1427   </learnerPreferenceData>
1428   <lessonStatus>browsed</lessonStatus>
1429   <location>Page 4, paragraph 6</location>
1430   <maxTimeAllowed>P2M3DT10H30M</maxTimeAllowed>
1431   <mode>browse</mode>
1432   <interactions>
1433     <interaction>
1434       <identifier>urn:ostyn.com:TFQ2003090412345</identifier>
1435       <objectiveIds>
1436         <objectiveId>1234-foobar-1234</objectiveId>
1437         <objectiveId>1235-barfoo-1345</objectiveId>
1438       </objectiveIds>
1439     </interaction>
  
```

```

1440     <description>Vanilla is lighter than chocolate. True or
1441 false?</description>
1442     <result>unanticipated</result>
1443     <latency>PT23.4S</latency>
1444     <weighting>.33</weighting>
1445     <type>>true_false</type>
1446     <correctResponses>
1447       <trueOrFalse>>true</trueOrFalse>
1448     </correctResponses>
1449     <learnerResponse/>
1450 </interaction>
1451 <interaction>
1452   <identifier>urn:ostyn.com:MCQ2003090412345</identifier>
1453   <objectiveIds>
1454     <objectiveId>
1455       1234-foobar-1234</objectiveId>
1456     <objectiveId>1235-barfoo-1345
1457   </objectiveId>
1458 </objectiveIds>
1459   <timeStamp>2005-10-17T09:32:47.45-05:00</timeStamp>
1460   <description>What was J.H.'s favorite ice cream
1461 combination?</description>
1462   <result>incorrect</result>
1463   <latency>PT23.4S</latency>
1464   <weighting>.33</weighting>
1465   <type>multiple_choice</type>
1466   <correctResponses>
1467     <choices>
1468       <choice>Vanilla</choice>
1469     </choices>
1470     <choices>
1471       <choice>Vanilla</choice>
1472     </choices>
1473     <choices>
1474       <choice>Chocolate</choice>
1475     </choices>
1476     <choices>
1477       <choice>Chocolate</choice>
1478       <choice>Vanilla</choice>
1479     </choices>
1480   </correctResponses>
1481   <learnerResponse>
1482     <choices>
1483       <choice>Pistachio</choice>
1484       <choice>Chocolate</choice>
1485     </choices>
1486   </learnerResponse>
1487 </interaction>
1488 <interaction>
1489   <identifier>urn:ostyn.com:FIQ2003090412345</identifier>
1490   <objectiveIds>
1491     <objectiveId>1234-foobar-1234</objectiveId>
1492     <objectiveId>1235-barfoo-1345</objectiveId>
1493   </objectiveIds>
1494   <timeStamp>2005-10-17T09:34:47-05:00</timeStamp>
1495   <description>Match things with numbers</description>
1496   <result>3.5926</result>
1497   <latency>PT30M</latency>
1498   <weighting>3</weighting>
1499   <type>fill_in</type>

```

```

1500     <correctResponses>
1501         <fillMatches caseMatters="false" orderMatters="false">
1502             <matchText lang="en">some</matchText>
1503             <matchText lang="en">thing</matchText>
1504         </fillMatches>
1505         <fillMatches caseMatters="false" orderMatters="true">
1506             <matchText>1</matchText>
1507             <matchText>2</matchText>
1508         </fillMatches>
1509     </correctResponses>
1510     <learnerResponse>
1511         <fillString lang="en">This is just any short response.</fillString>
1512     </learnerResponse>
1513 </interaction>
1514 <interaction>
1515     <identifier>urn:ostyn.com:LFQ2003090412345</identifier>
1516     <objectiveIds>
1517         <objectiveId>
1518             1234-foobar-1234</objectiveId>
1519         <objectiveId>
1520             1235-barfoo-1345
1521         </objectiveId>
1522     </objectiveIds>
1523     <timeStamp>2005-10-17T09:36:47-05:00</timeStamp>
1524     <description lang="fr">Début de la Ballade des Pendus</description>
1525     <result>-0.1415926</result>
1526     <latency>PT23.4S</latency>
1527     <weighting>33</weighting>
1528     <type>long_fill_in</type>
1529     <correctResponses>
1530         <matchText lang="fr" caseMatters="false">Frères humains qui après nous
1531 vivez,
1532 N'ayez les coeurs contre nous endurcis</matchText>
1533         <matchText lang="fr" caseMatters="false">Frères humains qui après nous
1534 vivez,
1535 N'ayez les cuers contre nous endurcis</matchText>
1536         <matchText lang="fr" caseMatters="true">Frères humains qui après nous
1537 vivez</matchText>
1538     </correctResponses>
1539     <learnerResponse>
1540         <longFillString lang="fr-BE">Ça commence avec "Frères
1541 humains..."</longFillString>
1542     </learnerResponse>
1543 </interaction>
1544 <interaction>
1545     <identifier>urn:ostyn.com:LIQ2003090412345</identifier>
1546     <objectiveIds>
1547         <objectiveId>
1548             1234-foobar-1234
1549         </objectiveId>
1550     </objectiveIds>
1551     <timeStamp>2005-10-17T09:38:47-05:00</timeStamp>
1552     <description>Which approach is most likely to succeed?</description>
1553     <latency>PT23.4S</latency>
1554     <weighting>.33</weighting>
1555     <type>likert</type>
1556     <learnerResponse>
1557         <choice>option_5</choice>
1558     </learnerResponse>
1559 </interaction>

```



```

1560     <interaction>
1561         <identifier>urn:ostyn.com:MAQ2003090412345</identifier>
1562         <objectiveIds>
1563             <objectiveId>
1564                 1234-foobar-1234</objectiveId>
1565             <objectiveId>
1566                 1235-barfoo-1345
1567             </objectiveId>
1568         </objectiveIds>
1569         <timeStamp>2005-10-17T09:40:47-05:00</timeStamp>
1570         <description>Connect the shmiblicks to the corresponding
1571 garfubles</description>
1572         <result>incorrect</result>
1573         <latency>PT23.4S</latency>
1574         <weighting>.33</weighting>
1575         <type>matching</type>
1576         <correctResponses>
1577             <matchPattern>
1578                 <pair>
1579                     <source>something_A</source>
1580                     <target>something_B</target>
1581                 </pair>
1582                 <pair>
1583                     <source>something_C</source>
1584                     <target>something_D</target>
1585                 </pair>
1586                 <pair>
1587                     <source>something_E</source>
1588                     <target>something_F</target>
1589                 </pair>
1590             </matchPattern>
1591             <matchPattern>
1592                 <pair>
1593                     <source>something_C</source>
1594                     <target>something_D</target>
1595                 </pair>
1596                 <pair>
1597                     <source>something_E</source>
1598                     <target>something_F</target>
1599                 </pair>
1600             </matchPattern>
1601         </correctResponses>
1602         <learnerResponse>
1603             <matchPattern>
1604                 <pair>
1605                     <source>something_C</source>
1606                     <target>something_D</target>
1607                 </pair>
1608                 <pair>
1609                     <source>something_E</source>
1610                     <target>something_F</target>
1611                 </pair>
1612             </matchPattern>
1613         </learnerResponse>
1614     </interaction>
1615     <interaction>
1616         <identifier>urn:ostyn.com:PEQ2003090412345</identifier>
1617         <objectiveIds>
1618             <objectiveId>
1619                 1234-foobar-1234</objectiveId>

```

```

1620     <objectiveId>
1621         1235-barfoo-1345
1622     </objectiveId>
1623 </objectiveIds>
1624     <timeStamp>2005-10-17T09:42:47-05:00</timeStamp>
1625     <description>Steps to diagnose the schbiblick</description>
1626     <result>3.5</result>
1627     <latency>PT23.4S</latency>
1628     <weighting>.33</weighting>
1629     <type>performance</type>
1630     <correctResponses>
1631         <performancePattern orderMatters="true">
1632             <step>
1633                 <stepName>StepB</stepName>
1634                 <stepAnswer>
1635                     <numeric min="10" max="10"/>
1636                 </stepAnswer>
1637             </step>
1638             <step>
1639                 <stepName>StepC</stepName>
1640                 <stepAnswer>
1641                     <literal>Green whatchamakalit</literal>
1642                 </stepAnswer>
1643             </step>
1644             <step>
1645                 <stepName>stepD</stepName>
1646             </step>
1647         </performancePattern>
1648         <performancePattern orderMatters="false">
1649             <step>
1650                 <stepName>stepA</stepName>
1651                 <stepAnswer>
1652                     <literal>Push the diagnostic button</literal>
1653                 </stepAnswer>
1654             </step>
1655             <step>
1656                 <stepName>stepE</stepName>
1657             </step>
1658         </performancePattern>
1659     </correctResponses>
1660     <learnerResponse>
1661         <step>
1662             <stepName>StepC</stepName>
1663             <stepAnswer>
1664                 <literal>Blue whatchamakalit</literal>
1665             </stepAnswer>
1666         </step>
1667         <step>
1668             <stepName>StepD</stepName>
1669             <stepAnswer>
1670                 <numeric>8.7</numeric>
1671             </stepAnswer>
1672         </step>
1673         <step>
1674             <stepName>StepD</stepName>
1675         </step>
1676     </learnerResponse>
1677 </interaction>
1678 <interaction>
1679     <identifier>urn:ostyn.com:INT2003090412345</identifier>

```

```

1680     <objectiveIds>
1681         <objectiveId>
1682             urn:bar.com/RCD/2345-800df-4%20test</objectiveId>
1683         <objectiveId>
1684             machinchose1234
1685         </objectiveId>
1686     </objectiveIds>
1687     <timeStamp>2005-10-17T09:44:47-05:00</timeStamp>
1688     <description>Steps to buy and enjoy ice cream.</description>
1689     <result>3.14159</result>
1690     <latency>PT23.4S</latency>
1691     <weighting>.33</weighting>
1692     <type>sequencing</type>
1693     <correctResponses>
1694         <stepSequence>
1695             <step>Choose_flavor</step>
1696             <step>Order_ice_cream</step>
1697             <step>Eat_ice_cream</step>
1698             <step>Wipe_chin</step>
1699         </stepSequence>
1700         <stepSequence>
1701             <step>Raid_fridge</step>
1702             <step>Choose_flavor</step>
1703             <step>Eat_ice_cream</step>
1704             <step>Wipe_chin</step>
1705         </stepSequence>
1706     </correctResponses>
1707     <learnerResponse>
1708         <steps>
1709             <step>Order_ice_cream</step>
1710             <step>Choose_flavor</step>
1711             <step>Eat_ice_cream</step>
1712             <step>Wipe_chin</step>
1713             <step>Eat_ice_cream</step>
1714             <step>Wipe_chin</step>
1715         </steps>
1716     </learnerResponse>
1717 </interaction>
1718 <interaction>
1719     <identifier>urn:ostyn.com:NUQ2003090412345</identifier>
1720     <objectiveIds>
1721         <objectiveId>
1722             1234-foobar-1234
1723         </objectiveId>
1724     </objectiveIds>
1725     <timeStamp>2005-10-17T09:17:47-05:00</timeStamp>
1726     <description>Pick a likely number for the result of this
1727 operation.</description>
1728     <result>3.14159</result>
1729     <latency>PT23.4S</latency>
1730     <weighting>7</weighting>
1731     <type>numeric</type>
1732     <correctResponses>
1733         <min>0</min>
1734         <max>123456783453.1415926</max>
1735     </correctResponses>
1736     <learnerResponse>
1737         <number>3.1415926</number>
1738     </learnerResponse>
1739 </interaction>

```

```

1740     <interaction>
1741         <identifier>urn:ostyn.com:OTQ2003090412345</identifier>
1742         <objectiveIds>
1743             <objectiveId>
1744                 1234-foobar-1234
1745             </objectiveId>
1746         </objectiveIds>
1747         <timeStamp>2005-10-17T09:13:47-05:00</timeStamp>
1748         <description>Some other kind of interaction</description>
1749         <result>correct</result>
1750         <latency>PT23.4S</latency>
1751         <weighting>.33</weighting>
1752         <type>other</type>
1753         <correctResponses>
1754             <correctOther>Something and <![CDATA[<Something>more or less
1755 complicated</Something>]]></correctOther>
1756         </correctResponses>
1757         <learnerResponse>
1758             <responseOther>Something more or less complicated</responseOther>
1759         </learnerResponse>
1760     </interaction>
1761 </interactions>
1762 <objectives>
1763     <objective>
1764         <identifier>urn:ostyn.com:id200309041234578</identifier>
1765         <completionStatus>incomplete</completionStatus>
1766         <description>Answer 10 questions</description>
1767         <score>
1768             <scaled>1.0</scaled>
1769             <max>77</max>
1770             <min>0</min>
1771             <raw>77</raw>
1772         </score>
1773         <status>browsed</status>
1774         <successStatus>failed</successStatus>
1775     </objective>
1776     <objective>
1777         <identifier>urn:ostyn.com:id200309041234576</identifier>
1778         <completionStatus>completed</completionStatus>
1779     </objective>
1780 </objectives>
1781 <progressMeasure>0.95</progressMeasure>
1782 <rawPassingScore>600</rawPassingScore>
1783 <scaledPassingScore>0.5</scaledPassingScore>
1784 <score>
1785     <scaled>0.5</scaled>
1786     <max>800</max>
1787     <min>400</min>
1788     <raw>600</raw>
1789 </score>
1790 <sessionTime>P3DT10H30M</sessionTime>
1791 <successStatus>passed</successStatus>
1792 <suspendData>Something=4; line break here:
1793 save this white space (10 spaces)[           ]end of line
1794 beginning of line. Something else on third line.</suspendData>
1795 <timeLimitAction>continue_message</timeLimitAction>
1796 <totalTime>P5DT10H30M</totalTime>
1797 </cocd>

```

1798

Figure C1—An example COCD XML instance

1799 **Annex D**

1800 (informative)

1801 **Explanatory XSD notes**

1802 This annex is a guide to the understanding and use of the XSD in Annex B. This annex refer-
1803 ences specific subclauses of Clause 6 of IEEE 1484.11.1–2004. IEEE 1484.11.1–2004 is re-
1804 quired to understand this annex.

1805 Where possible, this annex is organized in the same order as the subclauses of Clause 6 of IEEE
1806 1484.11.1–2004. The intent is to facilitate a parallel reading of this annex with IEEE 1484.11.1–
1807 2004, the XSD (see Annex B), and the sample XML instance (see Annex C).

1808 NOTES

1809 1—This annex is not intended to provide an explanation or rationale for every design or syntax choice
1810 implemented in the XSD nor is it intended to be an XML or XML Schema definition language tutorial.
1811 Familiarity with W3C XML Schema, Parts 1 and 2 is required to understand this annex. This annex men-
1812 tions features specified by the W3C XML Schema definition language and tested with current implemen-
1813 tations of XML schema processors.

1814 2—In all examples XML fragments in this annex, the `xs:` prefix denotes a type, element, group, or at-
1815 tribute name defined by the W3C namespace "http://www.w3.org/2001/XMLSchema".

1816 3—The first half of the XSD follows the order of the subclauses of Clause 6 of IEEE 1484.11.1–2004.
1817 The second half of the XSD uses conventional groupings of elements and type declarations: first ele-
1818 ments, then attributes, groups, simple types, and complex types. Within each grouping the elements are
1819 arranged alphabetically by name.

1820 **D.1 Limitations of the W3C XML Schema definition language for the** 1821 **representation of the data model**

1822 Some of the requirements defined in the data model cannot be expressed in the W3C XML
1823 Schema definition language in any way that would allow automatic validation or constraint en-
1824 forcement by a generic XML Schema processor. These requirements are discussed in D.1.1 –
1825 D.1.4.

1826 **D.1.1 Encoding of bags, arrays and sets**

1827 The bags, arrays, and sets defined by the data model are implemented with the XML Schema
1828 compositor `xs:sequence`. For collections of like elements, `xs:sequence` is used because it is
1829 the only way to allow a cardinality of more than one for the contained element, even though
1830 `xs:sequence` implies that the contents of the complex type have to appear as an ordered list. A
1831 typical XSD construct for a bag, array, or set is similar to the following example.

```
1832     <xs:element name="objectives">
1833         <xs:complexType>
1834             <xs:sequence>
1835                 <xs:element name="objective" type="objectiveType"
1836                     minOccurs="0" maxOccurs="unbounded">
1837                     </xs:element>
1838                 </xs:sequence>
1839             </xs:complexType>
1840     </xs:element>
```

1841 This may be instantiated in a COCD XML instance as in the following example.

```
1842     <objectives>
1843         <objective>
1844             <identifier>urn:foo.com:id200309041234578</identifier>
1845             <completionStatus>incomplete</completionStatus>
1846         </objective>
1847         <objective>
1848             <identifier>urn:foo.com:id200309041234534</identifier>
1849             <completionStatus>completed</completionStatus>
1850         </objective>
1851     </objectives>>
```

1852 The W3C XML Schema definition language cannot specify whether a collection implemented
1853 with `xs:sequence` should be treated as a bag (unordered) or an array (ordered). However, it is
1854 possible in some cases to specify whether the items in the collection have to have different con-
1855 tent, which is one of the characteristics of a set model. Implementers should refer to IEEE
1856 1484.11.1–2004 to determine whether a particular element defined as `xs:sequence` with a sin-
1857 gle element defined in the sequence should be treated as a bag, a set, or an array. As stated in
1858 W3C XML Schema Part 2, "The fact that this specification does not define an order-relation for
1859 some datatype does not mean that some other application cannot treat that datatype as being or-
1860 dered by imposing its own order relation."

1861 Whenever an `xs:sequence` construct is defined as part of a type definition, the XSD defines an
1862 attribute named `collectionType`. The value of this attribute is fixed as one of `bag`, `array`, or
1863 `set`. XML Schema syntax does not allow the addition of this attribute for element sequences
1864 defined in an `xs:group` construct. The `collectionType` attribute has no effect on XML vali-
1865 dation, but it may be useful for applications, because an XML Schema processor has to make the
1866 fixed attribute and its value available to applications even if they are not explicitly specified in
1867 the COCD XML instance. This Standard does not define any conformance requirements regard-
1868 ing the use or interpretation of the `collectionType` attribute.

1869 An implementation might use the fixed `collectionType` attribute as follows:

- 1870 – Instantiate an XML document that contains an element for which the `collectionType`
1871 attribute is defined in the XSD in an XML Schema processor environment.
- 1872 – Get the `collectionType` attribute and its value by calling usual XML Schema proces-
1873 sor methods. For example, using an XPath expression, such as `"./@collectionType"`,
1874 will access the attribute `collectionType` for the current element. The value can then
1875 be used to determine whether to treat the sequence of elements as ordered or unordered.

1876 D.1.2 Uniqueness

1877 Where possible, the standard XML Schema element `xs:unique` is used in the XSD to enforce
1878 uniqueness in collections that are defined as sets in the data model or that otherwise require
1879 uniqueness. These are

- 1880 – 6.1.9 Interactions—Each interaction requires an identifier, and the identifier has to be
1881 unique within the context of the content object.
- 1882 – 6.1.9.3 Objectives ID in an interaction record.
- 1883 – The choices in a set of choices in 6.1.9.5 Correct responses and 6.1.9.7 Learner re-
1884 sponse for multiple choice interactions.
- 1885 – 6.1.18 Objectives—Each objective requires an identifier, and the identifier has to be
1886 unique within the context of the content object.

1887 This allows an XML Schema processor to enforce the uniqueness constraint automatically. How-
1888 ever, validation of uniqueness for the other sets defined in the data model cannot be enforced
1889 automatically. It is not always possible to specify this constraint in the W3C XML Schema defi-
1890 nition language. For example, in the set of set of choices for the correct response for the interac-
1891 tion type `multiple_choice`, uniqueness can be specified for the inner set but not for the outer
1892 set. Enabling the constraint for the outer set would require adding considerable complexity, such
1893 as additional arbitrary identifiers for the elements of the outer set, and still would not guarantee
1894 uniqueness, because the actual content of the elements cannot be inspected. The outer set ele-
1895 ments would be guaranteed to have unique identifiers, but not that their content is unique.

1896 D.1.3 Smallest permitted maximums

1897 An SPM cannot be expressed in the W3C XML Schema definition language in any way that
1898 would allow automatic validation of an SPM constraint as defined in the data model. Therefore,
1899 the XSD always specifies `maxOccurs="unbounded"` when the data model allows multiplicity,
1900 and does not set a `maxLength` attribute for other types for which an SPM is specified in the data
1901 model. Implementers should refer to the IEEE 1484.11.1–2004 to determine the SPM that ap-
1902 plies to a particular element.

1903 Where possible, the XSD defines an attribute named `spm`. This attribute is added to several ele-
1904 ments and types defined in the XSD with a fixed value which is the SPM value. However, XML
1905 Schema syntax does not allow the addition of this attribute for element sequences defined in an
1906 `xs:group` construct nor is this attribute defined for string types that have to remain simple
1907 types, such as short and long identifiers. This attribute has no effect on XML validation, but it
1908 may be useful for applications, because an XML Schema processor has to make the fixed attrib-
1909 ute and its value available to applications even if they are not explicitly specified in the COCD
1910 XML instance. This Standard does not define any conformance requirements regarding the use
1911 or interpretation of the `spm` attribute.

1912 An implementation might use the fixed `spm` attribute as follows:

- 1913 – Instantiate an XML document that contains an element for which the `spm` attribute is de-
1914 fined in the XSD in an XML Schema processor environment.
- 1915 – Get the `spm` attribute and its value by calling usual XML Schema processor methods. For
1916 example, using an XPath expression, such as `"./@spm"`, will access the attribute `spm` for

1917 the current element. The attribute value can then be used to compare the length of the
1918 value in the node with the SPM.

1919 **D.1.4 Machine-readable annotations in the XSD**

1920 The W3C XML Schema definition language allows the inclusion of annotations that are intended
1921 to be machine readable in an XSD by encapsulating such annotation in an `xs:appInfo` element
1922 inside an `xs:annotation` element. The XSD contains annotations with information about re-
1923 quirements of the data model for particular elements.

1924 An implementation that is aware of this notational convention may be able to use an annotation
1925 to discover the SPM associated with an element when the XSD does not provide an SPM value
1926 as an `spm` attribute with a fixed value, or may be able to use the collection type when the XSD
1927 does not provide a `collectionType` attribute with a fixed value indicating the type of collec-
1928 tion. This Standard does not define any conformance requirements regarding annotations or the
1929 existence, use, or interpretation of the data expressed in `appInfo` elements in the XSD. (To
1930 make the examples in this annex more readable, the annotations have been removed from most
1931 of the XSD fragments.)

1932 The element names used within `appInfo` annotation elements are

- 1933 – `spm`: This name denotes the SPM value for the element.
- 1934 – `collectionType`: This name denotes the type of collection encoded using
1935 `xs:sequence`. The associated value is one of `bag`, `array`, or `set`.

1936 Ideally, such elements would be defined in some additional namespace specific to the data
1937 model, but creating such a namespace is outside of the scope of this Standard. Also, the name-
1938 space would have to be provided as a separate document, and the XSD would not be usable
1939 without the document. Referencing an external, arbitrary namespace would complicate deploy-
1940 ing and using the XSD. To avoid such complications but still allow a machine-readable represen-
1941 tation in the spirit of the W3C XML Schema recommendations, the data-model information data
1942 is "hidden" within a comment within an `appInfo` element. To discover the SPM or collection
1943 type for an element defined in the XSD an implementation can

- 1944 – Instantiate an XML fragment from the string contained in the `appInfo` element, if pre-
1945 sent, and extract the comment from that fragment;
- 1946 – Instantiate an XML fragment from the string contained in the comment; and
- 1947 – Get the element `spm` or `collectionType`, if present, and inspect its value.

1948 **D.2 Encoding of 6.1 Content object communication**

1949 The element specified in the data model as

```
1950 content_object_communication : record
```

1951 is represented in the XSD by the element `cocd` of type `cocdType`.

1952 A conforming COCD XML instance is expected to implement a `cocd` element as the root ele-
1953 ment of the XML document or fragment that contains a content object communication record.

1954 Every element in the complex type `cocdType` is optional and may appear at most one time in a
 1955 COCD XML instance. Although the elements are listed in the XSD in the same order as in sub-
 1956 clause 6.1 of IEEE 1484.11.1–2004, the elements may appear in any order.

```
1957     <xs:complexType name="cocdType">
1958         <xs:all>
1959             <xs:element ref="commentsFromLearner" minOccurs="0"/>
1960             <xs:element ref="commentsFromLMS" minOccurs="0"/>
1961             ...
1962             <xs:element ref="timeLimitAction" minOccurs="0"/>
1963             <xs:element ref="totalTime" minOccurs="0"/>
1964         </xs:all>
1965     </xs:complexType>
```

1966 Every element in the `cocdType` type definition references an element defined at the top level of
 1967 the XSD. These top-level elements appear in the XSD in the same order as in this type defini-
 1968 tion.

1969 **D.2.1 Encoding of 6.1.1 Comments from learner**

1970 This data-model element is implemented in the XSD as the element `commentsFromLearner`,
 1971 which is a sequence of zero or more `commentFromLearner` elements. Each
 1972 `commentFromLearner` element is of type `commentType`, which is an encoding of the com-
 1973 ment type defined in subclause 6.2.1 of IEEE 1484.11.1–2004.

```
1974     <xs:element name="commentsFromLearner">
1975         <xs:complexType>
1976             <xs:sequence>
1977                 <xs:element name="commentFromLearner" type="commentType"
1978                     minOccurs="0" maxOccurs="unbounded">
1979             </xs:sequence>
1980     </xs:element>
```

1981 **D.2.2 Encoding of 6.1.2 Comments from LMS**

1982 This data-model element is implemented in the XSD as the element `commentsFromLMS`, which
 1983 is a sequence of zero or more `commentFromLMS` elements. Each `commentFromLMS` element is
 1984 of type `commentType`, which is an implementation of the comment type defined in subclause
 1985 6.2.1 of IEEE 1484.11.1–2004.

```
1986     <xs:element name="commentsFromLMS">
1987         <xs:complexType>
1988             <xs:sequence>
1989                 <xs:element name="commentFromLMS" type="commentType"
1990                     minOccurs="0" maxOccurs="unbounded">
1991             </xs:sequence>
1992     </xs:element>
```

1993 **D.2.3 Encoding of 6.1.3 Completion status**

1994 This data-model element is implemented in the XSD as the element `completionStatus` of
 1995 type `completionStatusType`, which is an implementation of the completion status type de-
 1996 fined in subclause 6.2.2 of IEEE 1484.11.1–2004.

```
1997     <xs:element name="completionStatus" type="completionStatusType"/>
```

1998 **D.2.4 Encoding of 6.1.4 Completion threshold**

1999 This data-model element is implemented in the XSD as the element `completionThreshold` of
 2000 type `progressMeasureType`, which is an implementation of the progress measure type de-
 2001 fined in subclause 6.2.7 of IEEE 1484.11.1–2004.

```
2002     <xs:element name="completionThreshold" type="progressMeasureType"/>
```

2003 **D.2.5 Encoding of 6.1.5 Credit**

2004 This data-model element is implemented in the XSD as the element `credit`, which is defined as
 2005 an enumerated type with token values that correspond to the permissible values defined in sub-
 2006 clause 6.1.5 of IEEE 1484.11.1–2004.

```
2007     <xs:element name="credit">
2008         <xs:simpleType>
2009             <xs:restriction base="xs:token">
2010                 <xs:enumeration value="credit"/>
2011                 <xs:enumeration value="no_credit"/>
2012             </xs:restriction>
2013         </xs:simpleType>
2014     </xs:element>
```

2015 **D.2.6 Encoding of 6.1.6 Data model version**

2016 This data-model element is implemented in the XSD as the element `dataModelVersion` of
 2017 type `literalString250Type`, which is a custom type defined in the XSD to avoid accidental
 2018 modification of the string value if it contains white space. This type has the fixed attribute `spm`
 2019 with a value of 250.

```
2020     <xs:element name="dataModelVersion" type="literalString250Type"/>
```

2021 **D.2.7 Encoding of 6.1.7 Entry**

2022 This data-model element is implemented in the XSD as the element `entry`, which is defined as
 2023 an enumerated type with token values that correspond to the permissible values defined in sub-
 2024 clause 6.1.7 of IEEE 1484.11.1–2004.

```
2025     <xs:element name="entry">
2026         <xs:simpleType>
2027             <xs:restriction base="xs:token">
2028                 <xs:enumeration value="ab_initio"/>
2029                 <xs:enumeration value="resume"/>
2030                 <xs:enumeration value=""/>
2031             </xs:restriction>
2032         </xs:simpleType>
```

2033 </xs:element>

2034 NOTE—An empty string ("") is represented in a COCD XML instance as an empty element (e.g.,
2035 <cocd><entry/></cocd>).

2036 **D.2.8 Encoding of 6.1.8 Exit**

2037 This data-model element is implemented in the XSD as the element `exit`, which is defined as an
2038 enumerated type with token values that correspond to the permissible values defined in sub-
2039 clause 6.1.8 of IEEE 1484.11.1–2004.

```
2040     <xs:element name="exit">
2041       <xs:simpleType>
2042         <xs:restriction base="xs:token">
2043           <xs:enumeration value="logout"/>
2044           <xs:enumeration value="normal"/>
2045           <xs:enumeration value="suspend"/>
2046           <xs:enumeration value="timeout"/>
2047           <xs:enumeration value=""/>
2048         </xs:restriction>
2049       </xs:simpleType>
2050     </xs:element>
```

2051 NOTE—An empty string ("") is represented in a COCD XML instance as an empty element (e.g.,
2052 <cocd><exit/></cocd>).

2053 **D.2.9 Encoding of 6.1.9 Interactions**

2054 This data model element is implemented in the XSD as a sequence of elements of type
2055 `interactionType`. The type `interactionsType` is defined as a collection of `interaction`
2056 elements, each of which represents an interaction record. Although the data model specifies that
2057 the collection is a bag of `interaction_type` records, the XSD has to use `xs:sequence` to
2058 allow multiplicity of the `interaction` element. In this case, `xs:sequence` should not be in-
2059 terpreted as implying any particular order.

```
2060     <xs:complexType name="interactionsType">
2061       <xs:sequence>
2062         <xs:element name="interaction" type="interactionType"
2063           minOccurs="0" maxOccurs="unbounded"/>
2064       </xs:sequence>
2065     </xs:complexType>
```

2066 **Implementation of `interaction_type`**

2067 This data-model structure is implemented in the XSD by the global type `interactionType`,
2068 which includes the required elements `identifier` and `type` and the optional elements
2069 `objectiveIds`, `timeStamp`, `correctResponses`, `weighting`, `learnerResponse`,
2070 `result`, `latency`, and `description`. If present, these elements may occur in any order. The
2071 elements `correctResponses` and `learnerResponse` are defined as global elements to facili-
2072 tate possible reuse.

```
2073     <xs:complexType name="interactionType">
2074       <xs:all>
```

```

2075     <xs:element name="identifier" type="longIdentifierType">
2076     <xs:element name="type" type="interactionTypeType"/>
2077     <xs:element name="objectiveIds" type="objectiveIdsType"
2078         minOccurs="0">
2079         <xs:unique name="uniqueInObjectivesIds">
2080             <xs:selector xpath="."/>
2081             <xs:field xpath="./objectiveId"/>
2082         </xs:unique>
2083     </xs:element>
2084     <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
2085     <xs:element name="correctResponses" type="correctResponsesType"
2086         minOccurs="0"/>
2087     <xs:element name="weighting" type="real7Type" minOccurs="0"/>
2088     <xs:element name="learnerResponse type="learnerResponseType"
2089         minOccurs="0"/>
2090     <xs:element name="result" type="interactionResultType"
2091         minOccurs="0"/>
2092     <xs:element name="latency" type="timeIntervalType"
2093         minOccurs="0"/>
2094     <xs:element name="description" type="localizedString250Type"
2095         minOccurs="0"/>
2096 </xs:all>
2097 </xs:complexType>

```

2098 **Encoding of 6.1.9.1 ID**

2099 This data-model element is implemented in the XSD as the element identifier of type
2100 longIdentifierType.

```
2101     <xs:element name="identifier" type="longIdentifierType"/>
```

2102 **Encoding of 6.1.9.2 Type**

2103 This data-model element is implemented in the XSD as the element type of type
2104 interactionTypeType.

```
2105     <xs:element name="type" type="interactionTypeType"/>
```

2106 **Interaction type type**

2107 This data-model structure is implemented in the XSD as the global type
2108 interactionTypeType, which is defined as an enumerated type with token values that corre-
2109 spond to the permissible values defined in subclause 6.1.9.2 of IEEE 1484.11.1–2004.

```

2110     <xs:simpleType name="interactionTypeType">
2111         <xs:restriction base="xs:token">
2112             <xs:enumeration value="true_false"/>
2113             <xs:enumeration value="multiple_choice"/>
2114             <xs:enumeration value="fill_in"/>
2115             <xs:enumeration value="long_fill_in"/>
2116             <xs:enumeration value="likert"/>
2117             <xs:enumeration value="matching"/>
2118             <xs:enumeration value="performance"/>
2119             <xs:enumeration value="sequencing"/>

```

```
2120     <xs:enumeration value="numeric"/>
2121     <xs:enumeration value="other"/>
2122 </xs:restriction>
2123 </xs:simpleType>
```

2124 **Encoding of 6.1.9.3 Objectives ID**

2125 This data-model element is implemented in XSD as the element `ObjectiveIds` of type
2126 `objectiveIdsType`, which includes zero or more instances of the element
2127 `longIdentifierType`. A uniqueness constraint is applied.

```
2128     <xs:element name="objectiveIds" type="objectiveIdsType"
2129       minOccurs="0">
2130       <xs:unique name="uniqueInObjectivesIds">
2131         <xs:selector xpath="./objectiveId"/>
2132         <xs:field xpath="."/>
2133       </xs:unique>
2134     </xs:element>
```

2135 The global type `objectiveIdsType` is defined as a collection of `objectiveId` elements of
2136 type `longIdentifierType`.

```
2137     <xs:complexType name="objectiveIdsType">
2138       <xs:sequence>
2139         <xs:element name="objectiveId" type="longIdentifierType"
2140           minOccurs="0" maxOccurs="unbounded"/>
2141       </xs:sequence>
2142     </xs:complexType>
```

2143 **Encoding of 6.1.9.4 Time stamp**

2144 This data-model element is implemented in the XSD as the element `timestamp` of type
2145 `dateTimeType`.

```
2146     <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
```

2147 **Encoding of 6.1.9.5 Correct responses**

2148 This data-model element is implemented in the XSD as the element `correctResponses` of
2149 type `correctResponsesType`.

```
2150     <xs:element name="correctResponses" type="correctResponsesType"
2151       minOccurs="0"/>
```

2152 **Correct responses type**

2153 This global type implements the various `correct_responses` data structures defined in the
 2154 data model to correspond to different types of interactions. Regardless of the complexity of the
 2155 data structure for a type of interaction, the choice is implemented in the XSD by a global group.
 2156 It is up to the COCD XML instance implementation to choose the correct group to match the in-
 2157 teraction type. This correspondence cannot be expressed or validated using the W3C XML
 2158 Schema definition language. Refer to the sample COCD XML instance (see Annex C) for exam-
 2159 ples showing how this type can be used in an actual COCD XML instance.

```
2160 <xs:complexType name="correctResponsesType">
2161   <xs:choice>
2162     <xs:group ref="grpCorrectTrueFalse"/>
2163     <xs:group ref="grpCorrectMultipleChoice"/>
2164     <xs:group ref="grpCorrectFillIn"/>
2165     <xs:group ref="grpCorrectLongFillIn"/>
2166     <xs:group ref="grpCorrectLikert"/>
2167     <xs:group ref="grpCorrectMatching"/>
2168     <xs:group ref="grpCorrectPerformance"/>
2169     <xs:group ref="grpCorrectSequencing"/>
2170     <xs:group ref="grpCorrectNumeric"/>
2171     <xs:group ref="grpCorrectOther"/>
2172   </xs:choice>
2173 </xs:complexType>
```

2174 These groups are described in detail in D.6.

2175 **Encoding of 6.1.9.6 Weighting**

2176 This data-model element is implemented in the XSD as the element `weighting` of type
 2177 `real7Type`.

```
2178 <xs:element name="weighting" type="real7Type" minOccurs="0"/>
```

2179 **Encoding of 6.1.9.7 Learner response**

2180 This data-model element is implemented in the XSD as the element `learnerResponse` of type
 2181 `learnerResponseType`.

```
2182 <xs:element name="learnerResponse" type="learnerResponseType"
2183   minOccurs="0"/>
```

2184 **Learner response type**

2185 This global type implements the various `learner_response` data structures defined in the data
 2186 model to correspond to different types of interaction types. Regardless of the complexity of the
 2187 data structure for a type of interaction, the choice is implemented in the XSD as a global group.
 2188 It is up to the COCD XML instance implementation to choose the correct group to match the in-
 2189 teraction type. Refer to the sample COCD XML instance (see Annex C) for examples of how
 2190 this type can typically be used in an actual COCD XML instance.

```
2191 <xs:complexType name="learnerResponseType">
2192   <xs:choice>
```

```

2193     <xs:group ref="grpResponseTrueFalse"/>
2194     <xs:group ref="grpResponseMultipleChoice"/>
2195     <xs:group ref="grpResponseFillIn"/>
2196     <xs:group ref="grpResponseLongFillIn"/>
2197     <xs:group ref="grpResponseLikert"/>
2198     <xs:group ref="grpResponseMatching"/>
2199     <xs:group ref="grpResponsePerformance"/>
2200     <xs:group ref="grpResponseSequencing"/>
2201     <xs:group ref="grpResponseNumeric"/>
2202     <xs:group ref="grpResponseOther"/>
2203 </xs:choice>
2204 </xs:complexType>

```

2205 These groups are described in detail in D.6.

2206 **Encoding of 6.1.9.8 Result**

2207 This data-model element is implemented in the XSD as the element `result` of type
2208 `interactionResultType`.

```

2209     <xs:element name="result" type="interactionResultType"
2210       minOccurs="0"/>

```

2211 **Interaction result type**

2212 This global type implements the data-model requirement that the value of `result` may be a to-
2213 ken that corresponds to a permissible value defined by the data model or a numeric value. This
2214 element uses `xs:union` instead of `xs:choice` to avoid having to define subelements with arbi-
2215 trary names in the XSD, which would force another layer of nesting. The `xs:union` construct
2216 allows only a single value to occur, and thus meets the requirement of the data model.

```

2217     <xs:simpleType name="interactionResultType">
2218       <xs:union memberTypes="real7Type interactionResultTokenType"/>
2219     </xs:simpleType>
2220     <xs:simpleType name="interactionResultTokenType">
2221       <xs:restriction base="xs:token">
2222         <xs:enumeration value="correct"/>
2223         <xs:enumeration value="incorrect"/>
2224         <xs:enumeration value="neutral"/>
2225         <xs:enumeration value="unanticipated"/>
2226       </xs:restriction>
2227     </xs:simpleType>

```

2228 **Encoding of 6.1.9.9 Latency**

2229 This data-model element is implemented in the XSD as the element `latency` of type
2230 `timeIntervalType`.

```

2231     <xs:element name="latency" type="timeIntervalType" minOccurs="0"/>

```


2232 **Encoding of 6.1.9.10 Description**

2233 This data-model element is implemented in the XSD as the element description of type
2234 localizedString250Type.

```
2235     <xs:element name="description" type="localizedString250Type"  
2236         minOccurs="0" />
```

2237 **D.2.10 Encoding of 6.1.10 Launch data**

2238 This data-model element is implemented in the XSD as the element launchData of type
2239 literalString4000Type to avoid accidental modification of the string value if it contains
2240 white space.

```
2241     <xs:element name="launchData" type="literalString4000Type" />
```

2242 **D.2.11 Encoding of 6.1.11 Learner ID**

2243 This data-model element is implemented in the XSD as the element learnerId of type
2244 longIdentifierType.

```
2245     <xs:element name="learnerId" type="longIdentifierType" />
```

2246 **D.2.12 Encoding of 6.1.12 Learner name**

2247 This data-model element is implemented in the XSD as the element learnerName of type
2248 localizedString250Type.

```
2249     <xs:element name="learnerName" type="localizedString250Type" />
```

2250 **D.2.13 Encoding of 6.1.13 Learner preference data**

2251 This data-model element is implemented in the XSD as the element learnerPreferenceData
2252 of type learnerPreferenceType.

```
2253     <xs:element name="learnerPreferenceData"  
2254         type="learnerPreferenceType" />
```

2255 The global type learnerPreferenceType implements the learner_preference_type
2256 defined in subclause 6.1.13 of IEEE 1484.11.1–2004. It contains four optional elements that may
2257 appear in any order, audioLevel, language, deliverySpeed, and audioCaptioning,
2258 which are described in more detail below.

```
2259     <xs:element name="learnerPreferenceData">  
2260         <xs:complexType>  
2261             <xs:all>  
2262                 <xs:element name="audioLevel" minOccurs="0">  
2263                     ...  
2264                 </xs:element>  
2265                 <xs:element name="language" type="languageType"  
2266                     minOccurs="0" />  
2267                 <xs:element name="deliverySpeed" minOccurs="0">  
2268                     ...  
2269                 </xs:element>  
2270                 <xs:element name="audioCaptioning" minOccurs="0">
```

```

2271         ...
2272         </xs:element>
2273     </xs:all>
2274 </xs:complexType>
2275 </xs:element>

```

2276 **Encoding of 6.1.13.1 Audio level**

2277 This data-model element is implemented in the XSD as the element `audioLevel`, which is de-
 2278 fined inline as an optional element of `learnerPreferenceData`. It is based on the type
 2279 `real7Type`, with a restriction that its value is greater than or equal to zero.

```

2280     <xs:element name="audioLevel" minOccurs="0">
2281         <xs:simpleType>
2282             <xs:restriction base="real7Type">
2283                 <xs:minInclusive value="0"/>
2284             </xs:restriction>
2285         </xs:simpleType>
2286     </xs:element>

```

2287 **Encoding of 6.1.13.2 Language**

2288 This data-model element is implemented in the XSD as the element `language`, which is defined
 2289 inline as an optional element of `learnerPreferenceData`. It is of type `languageType`.

```

2290     <xs:element name="language" type="languageType" minOccurs="0"/>

```

2291 **Encoding of 6.1.13.3 Delivery speed**

2292 This data-model element is implemented in the XSD as the element `deliverySpeed`, which is
 2293 defined inline as an optional element of `learnerPreferenceData`. It is based on the type
 2294 `real7Type`, with a restriction that its value is greater than or equal to zero.

```

2295     <xs:element name="deliverySpeed" minOccurs="0">
2296         <xs:simpleType>
2297             <xs:restriction base="real7Type">
2298                 <xs:minInclusive value="0"/>
2299             </xs:restriction>
2300         </xs:simpleType>
2301     </xs:element>

```

2302 **Encoding of 6.1.13.4 Audio captioning**

2303 This data-model element is implemented in the XSD as the element `audioCaptioning`, which
 2304 is defined inline as an optional element of `learnerPreferenceData`. It is defined as an enu-
 2305 merated type with token values that correspond to the permissible values defined in subclause
 2306 6.1.13.4 of IEEE 1484.11.1–2004.

```

2307     <xs:element name="audioCaptioning" minOccurs="0">
2308         <xs:simpleType>
2309             <xs:restriction base="xs:token">
2310                 <xs:enumeration value="off"/>
2311                 <xs:enumeration value="no_change"/>
2312                 <xs:enumeration value="on"/>

```

```
2313         </xs:restriction>
2314     </xs:simpleType>
2315 </xs:element>
```

2316 **D.2.14 Encoding of 6.1.14 Lesson status**

2317 This data-model element is implemented in the XSD as the element `lessonStatus` of type
2318 `legacyStatusType`.

```
2319     <xs:element name="lessonStatus" type="legacyStatusType"/>
```

2320 **D.2.15 Encoding of 6.1.15 Location**

2321 This data-model element is implemented in the XSD as the element `location` of type
2322 `literalString1000Type` to avoid accidental modification of the string value if it contains
2323 white space.

```
2324     <xs:element name="location" type="literalString1000Type"/>
```

2325 **D.2.16 Encoding of 6.1.16 Max time allowed**

2326 This data-model element is implemented in the XSD as the element `maxTimeAllowed` of type
2327 `timeIntervalType`.

```
2328     <xs:element name="maxTimeAllowed" type="timeIntervalType"/>
```

2329 **D.2.17 Encoding of 6.1.17 Mode**

2330 This data-model element is implemented in the XSD as the element `mode`, which is defined as an
2331 enumerated type with token values that correspond to the permissible values defined in sub-
2332 clause 6.1.17 of IEEE 1484.11.1–2004.

```
2333     <xs:element name="mode">
2334         <xs:simpleType>
2335             <xs:restriction base="xs:token">
2336                 <xs:enumeration value="browse"/>
2337                 <xs:enumeration value="normal"/>
2338                 <xs:enumeration value="review"/>
2339             </xs:restriction>
2340         </xs:simpleType>
2341     </xs:element>
```

2342 **D.2.18 Encoding of 6.1.18 Objectives**

2343 This data-model element is implemented in the XSD as a sequence of objective elements of
2344 type `objectiveType`. Although the data model specifies that this is a set of objective, the
2345 XSD has to use `xs:sequence` to allow multiplicity of the objective element. In this case,
2346 `xs:sequence` should not be interpreted as implying any particular order. The value of the man-
2347 datory identifier element within each objective element in this collection has to be
2348 unique.

```
2349     <xs:element name="objectives" type="objectivesType">
2350         <xs:unique name="uniqueInSetOfObjectives">
```

```

2351         <xs:selector xpath="./objective"/>
2352         <xs:field xpath="identifier"/>
2353     </xs:unique>
2354 </xs:element>

```

2355 The global type `objectivesType` implements a collection of objective elements of type
 2356 `objectiveType`.

```

2357     <xs:complexType name="objectivesType">
2358         <xs:sequence>
2359             <xs:element name="objective" type="objectiveType" minOccurs="0"
2360                 maxOccurs="unbounded">
2361             </xs:element>
2362         </xs:sequence>
2363         <xs:attribute name="spm" fixed="100"/>
2364     </xs:complexType>

```

2365 Objective type

2366 This data-model structure is implemented in the XSD as the global type `objectiveType`,
 2367 which includes the required element `identifier` and the optional elements `score`, `status`,
 2368 `progressMeasure`, `completionStatus`, `successStatus`, and `description`. If present,
 2369 these elements may occur in any order.

```

2370     <xs:complexType name="objectiveType">
2371         <xs:all>
2372             <xs:element name="identifier" type="longIdentifierType"/>
2373             <xs:element name="score" type="scoreType" minOccurs="0"/>
2374             <xs:element name="status" type="legacyStatusType"
2375                 minOccurs="0"/>
2376             <xs:element name="progressMeasure"
2377                 type="progressMeasureType" minOccurs="0"/>
2378             <xs:element name="completionStatus"
2379                 type="completionStatusType" minOccurs="0"/>
2380             <xs:element name="successStatus" type="successStatusType"
2381                 minOccurs="0"/>
2382             <xs:element name="description" type="localizedString250Type"
2383                 minOccurs="0">
2384             </xs:element>
2385         </xs:all>
2386     </xs:complexType>

```

2387 D.2.19 Encoding of 6.1.19 Progress measure

2388 This data-model element is implemented in the XSD as the element `progressMeasure` of type
 2389 `progressMeasureType`.

```

2390     <xs:element name="progressMeasure" type="progressMeasureType"/>

```

2391 D.2.20 Encoding of 6.1.20 Raw passing score

2392 This data-model element is implemented in the XSD as the element `rawPassingScore` of type
 2393 `real7Type`.

```

2394     <xs:element name="rawPassingScore" type="real7Type"/>

```

2395 **D.2.21 Encoding of 6.1.21 Scaled passing score**

2396 This data-model element is implemented in the XSD as the element `scaledPassingScore` of
2397 type `scaledScoreType`.

```
2398     <xs:element name="scaledPassingScore" type="scaledScoreType"/>
```

2399 **D.2.22 Encoding of 6.1.22 Score**

2400 This data-model element is implemented in the XSD as the element `score` of type `scoreType`,
2401 which is a complex type composed of several elements.

```
2402     <xs:element name="score" type="scoreType"/>
```

2403 **D.2.23 Encoding of 6.1.23 Session time**

2404 This data-model element is implemented in the XSD as the element `sessionTime` of type
2405 `timeIntervalType`.

```
2406     <xs:element name="sessionTime" type="timeIntervalType"/>
```

2407 **D.2.24 Encoding of 6.1.24 Success status**

2408 This data-model element is implemented in the XSD as the element `successStatus` of type
2409 `successStatusType`.

```
2410     <xs:element name="successStatus" type="successStatusType"/>
```

2411 **D.2.25 Encoding of 6.1.25 Suspend data**

2412 This data-model element is implemented in the XSD as the element `suspendData` of type
2413 `literalString4000Type` to avoid accidental modification of the string value if it contains
2414 white space.

```
2415     <xs:element name="suspendData" type="literalString4000Type"/>
```

2416 **D.2.26 Encoding of 6.1.26 Time limit action**

2417 This data-model element is implemented in the XSD as the element `timeLimitAction`, which
2418 is defined as an enumerated type with token values that correspond to the permissible values de-
2419 fined in subclause 6.1.26 of IEEE 1484.11.1–2004.

```
2420     <xs:element name="timeLimitAction">
2421       <xs:simpleType>
2422         <xs:restriction base="xs:token">
2423           <xs:enumeration value="continue_message"/>
2424           <xs:enumeration value="continue_no_message"/>
2425           <xs:enumeration value="exit_message"/>
2426           <xs:enumeration value="exit_no_message"/>
2427         </xs:restriction>
2428       </xs:simpleType>
2429     </xs:element>
```

2430 **D.2.27 Encoding of 6.1.27 Total time**

2431 This data-model element is implemented in the XSD as the element `totalTime` of type
2432 `timeIntervalType`.

```
2433 <xs:element name="totalTime" type="timeIntervalType"/>
```

2434 **D.3 Encoding of 6.2 Auxiliary data types**

2435 The encodings of the data-model auxiliary data types are discussed in D.3.1 – D.3.10.

2436 **D.3.1 Implementation of 6.2.1 Comment type**

2437 This data-model type is implemented in the XSD as the global type `commentType`, which in-
2438 cludes the required element `comment` and the optional elements `location` and `timeStamp`. If
2439 present, these elements may occur in any order.

```
2440 <xs:complexType name="commentType">
2441 <xs:all>
2442 <xs:element name="comment" type="localizedString4000Type"/>
2443 <xs:element name="location" type="literalString1000Type"
2444 minOccurs="0"/>
2445 <xs:element name="timeStamp" type="dateTimeType" minOccurs="0"/>
2446 </xs:all>
2447 </xs:complexType>
```

2448 **D.3.2 Implementation of 6.2.2 Completion status type**

2449 This data-model type is implemented in the XSD as the global type `completionStatusType`,
2450 which is defined as an enumerated type with token values that correspond to the permissible val-
2451 ues defined in subclause 6.2.2 of IEEE 1484.11.1–2004.

```
2452 <xs:simpleType name="completionStatusType">
2453 <xs:restriction base="xs:token">
2454 <xs:enumeration value="completed"/>
2455 <xs:enumeration value="incomplete"/>
2456 <xs:enumeration value="not_attempted"/>
2457 <xs:enumeration value="unknown"/>
2458 </xs:restriction>
2459 </xs:simpleType>
```

2460 **D.3.3 Implementation of 6.2.3 Date time type**

2461 This data-model type is implemented in the XSD as the global type `dateTimeType`, which is
2462 based on the XML Schema primitive data type `dateTime`. The `dateTime` type is a conforming
2463 implementation of the requirements specified defined in subclause 6.2.3 of IEEE 1484.11.1–
2464 2004.

```
2465 <xs:simpleType name="dateTimeType">
2466 <xs:restriction base="xs:dateTime"/>
2467 </xs:simpleType>
```

2468 The format for `dateTime` is defined by the following pattern:

2469 YYYY[-MM[-DD[Thh[:mm[:ss[.s]]][TZD]]]]

2470 where anything enclosed in square brackets is optional.

2471 **D.3.4 Implementation of 6.2.4 Language type**

2472 This data-model type is implemented in the XSD as the global type `languageType`, which is
 2473 based on the XML Schema derived data type `language`. The `language` type is a conforming
 2474 implementation of the requirements specified defined in subclause 6.2.4 of IEEE 1484.11.1–
 2475 2004 except for the specification of an SPM, which is represented in an annotation in the XSD.

```
2476       <xs:simpleType name="languageType">
2477        <xs:restriction base="xs:language"/>
2478       </xs:simpleType>
```

2479 **D.3.5 Implementation of 6.2.5 Localized string type**

2480 This data-model type is implemented in the XSD as the global type `localizedStringType`,
 2481 which is based on the global type `literalStringType`. That base type is used to avoid acci-
 2482 dental modification of the string value if it contains white space. The type
 2483 `localizedStringType` is used as an abstract base type for two other types,
 2484 `localizedString250Type` and `localizedString4000Type`, each of which has a different
 2485 fixed value for an `spm` attribute.

```
2486       <xs:complexType name="localizedStringType" abstract="true">
2487        <xs:simpleContent>
2488         <xs:extension base="literalStringType">
2489         <xs:attribute name="lang" type="languageType"/>
2490         </xs:extension>
2491        </xs:simpleContent>
2492       </xs:complexType>
```

2493 **D.3.6 Implementation of 6.2.6 Long identifier type**

2494 This data-model type is implemented in the XSD as the global type `longIdentifierType`,
 2495 which is based on the XML Schema primitive data type `anyURI`. The `anyURI` type is a con-
 2496 forming implementation of the requirements defined in subclause 6.2.6 of IEEE 1484.11.1–2004
 2497 except for the specification of an SPM, which is represented in an annotation in the XSD. Be-
 2498 cause this type is used where a simple type is required, no `spm` attribute can be attached to it.

```
2499       <xs:simpleType name="longIdentifierType">
2500        <xs:restriction base="xs:anyURI">
2501        </xs:restriction>
2502       </xs:simpleType>
```

2503 **D.3.7 Implementation of 6.2.7 Progress measure type**

2504 This data-model type is implemented in the XSD as the global type `progressMeasureType`,
 2505 which is based on the global type `real7Type`, which, in turn, is based on the XML Schema
 2506 primitive data type `decimal`.

```
2507       <xs:simpleType name="progressMeasureType">
```

```
2508     <xs:restriction base="real7Type">
2509         <xs:minInclusive value="0"/>
2510         <xs:maxInclusive value="1"/>
2511     </xs:restriction>
2512 </xs:simpleType>
```

2513 **D.3.8 Implementation of 6.2.8 Score type**

2514 This data-model type is implemented in the XSD as the global type `scoreType`, which includes
2515 the optional elements `scaled`, `max`, `min`, and `raw`. If present, these elements may occur in any
2516 order.

```
2517     <xs:complexType name="scoreType">
2518         <xs:all>
2519             <xs:element name="scaled" type="scaledScoreType"/>
2520             <xs:element name="max" type="real7Type" minOccurs="0"/>
2521             <xs:element name="min" type="real7Type" minOccurs="0"/>
2522             <xs:element name="raw" type="real7Type" minOccurs="0"/>
2523         </xs:all>
2524     </xs:complexType>
```

2525 **D.3.9 Implementation of 6.2.9 Short identifier type**

2526 This data-model type is implemented in the XSD as the global type `shortIdentifierType`,
2527 which is based on the XML Schema primitive data type `anyURI`. The `anyURI` type is a con-
2528 forming implementation of the requirements defined in subclause 6.2.9 of IEEE 1484.11.1–2004
2529 except for the specification of an SPM, which is represented in an annotation in the XSD. Be-
2530 cause this type is used where a simple type is required, no `spm` attribute can be attached to it.

```
2531     <xs:simpleType name="shortIdentifierType">
2532         <xs:restriction base="xs:anyURI">
2533     </xs:simpleType>
```

2534 **D.3.10 Implementation of 6.2.10 Success status type**

2535 This data-model type is implemented in the XSD as the global type `successStatusType`
2536 which is defined as an enumerated type with token values that correspond to the permissible val-
2537 ues defined in subclause 6.2.10 of IEEE 1484.11.1–2004.

```
2538     <xs:simpleType name="successStatusType">
2539         <xs:restriction base="xs:token">
2540             <xs:enumeration value="failed"/>
2541             <xs:enumeration value="passed"/>
2542             <xs:enumeration value="unknown"/>
2543         </xs:restriction>
2544     </xs:simpleType>
```

2545 **D.4 Implementation of other documented data types**

2546 The implementations of other documented data types are discussed in D.4.1 – D.4.2.

2547 **D.4.1 Implementation of real(10,7)**

2548 An explanation of this type is provided in subclause B.1 of IEEE 1484.11.1–2004. This data-
 2549 model type is implemented in the XSD as the global type `real7Type`, which is based on the
 2550 XML Schema primitive data type `decimal`.

```
2551     <xs:simpleType name="real7Type">
2552         <xs:restriction base="xs:decimal"/>
2553     </xs:simpleType>
```

2554 By neither restricting the number of fraction digits nor the number of digits in total, the XSD
 2555 provides the required precision in all cases of practical interest.

2556 **D.4.2 Implementation of the time interval data type**

2557 An explanation of this type is provided in subclause B.2 of IEEE 1484.11.1–2004. This data-
 2558 model type is implemented in the XSD as the global type `timeIntervalType`, which is based
 2559 on the XML Schema primitive data type `duration`. The `duration` type is a conforming im-
 2560 plementation of the requirements specified in the data model.

```
2561     <xs:simpleType name="timeIntervalType">
2562         <xs:restriction base="xs:duration"/>
2563     </xs:simpleType>
```

2564 This type definition in the XSD does not enforce a restriction on the number of decimal digits for
 2565 the seconds part of the duration expression.

2566 **D.5 Other global types defined in the XSD**

2567 The types in D.5.1 – D.5.3 are used by more than one element in the XSD or as building blocks
 2568 in the definition of other XSD types.

2569 **D.5.1 Literal string type**

2570 This type is defined for strings in which white space should not be modified by an XML imple-
 2571 mentation. The type `literalStringType` is used as an abstract base type for three other types,
 2572 `literalString250Type`, `literalString1000Type`, and `literalString4000Type`,
 2573 each of which has a different fixed value for a `spm` attribute.

```
2574     <xs:simpleType name="literalStringType">
2575         <xs:restriction base="xs:string">
2576             <xs:whiteSpace value="preserve"/>
2577         </xs:restriction>
2578     </xs:simpleType>
```

2579 **D.5.2 Lesson status type**

2580 This data-model type is defined as the global type `obsoleteStatusType` which is defined as
2581 an enumerated type with token values that correspond to the permissible values defined in sub-
2582 clause 6.1.14 of IEEE 1484.11.1–2004.

```
2583     <xs:simpleType name="obsoleteStatusType">  
2584         <xs:restriction base="xs:token">  
2585             <xs:enumeration value="browsed"/>  
2586             <xs:enumeration value="completed"/>  
2587             <xs:enumeration value="failed"/>  
2588             <xs:enumeration value="incomplete"/>  
2589             <xs:enumeration value="not_attempted"/>  
2590             <xs:enumeration value="passed"/>  
2591         </xs:restriction>  
2592     </xs:simpleType>
```

2593 **D.5.3 Scaled score type**

2594 This data-model type is defined as a numeric type based on `real7Type` with the range con-
2595 straints defined in subclause 6.1.21 of IEEE 1484.11.1–2004.

```
2596     <xs:simpleType name="scaledScoreType">  
2597         <xs:restriction base="real7Type">  
2598             <xs:minInclusive value="-1"/>  
2599             <xs:maxInclusive value="1"/>  
2600         </xs:restriction>  
2601     </xs:simpleType>
```

2602 **D.5.4 Literal string type – XML specific**

2603 This type is not defined explicitly in IEEE 1484.11.1–2004, but it is implicit in references to the
2604 ISO 11404 [B2] characterstring data type. Because XML Schema processors may modify the
2605 white space in a string value, it is necessary to specify in the XSD that the values for various
2606 string-based elements defined in subclause 6.1. of IEEE 1484.11.1–2004 cannot be modified.
2607 This is done by defining the type `literalStringType` that specifies that white space must be
2608 preserved. This type is used as the base type to define the type `localizedStringType` as well
2609 as the three types `literalString250Type`, `literalString1000Type` and
2610 `literalString4000Type` with fixed `spm` attribute values of 250, 1000 and 4000, re-
2611 spectively.

```
2612     <xs:simpleType name="literalStringType">  
2613         <xs:restriction base="xs:string">  
2614             <xs:whiteSpace value="preserve"/>  
2615         </xs:restriction>  
2616     </xs:simpleType>
```

2617 **D.6 Elements and groups used to implement response data**

2618 The globally defined, reusable, XML elements and groups discussed in D.6.1 – D.6.21 are used
 2619 in the XSD as parts of the definitions for the complex, data-model types for correct responses
 2620 and learner response. They are listed in alphabetic order. These XSD fragments are easier to un-
 2621 derstand by looking at the sample COCD XML instance (see Annex C).

2622 **D.6.1 Correct responses for fill-in**

2623 In this group definition, each `fillMatches` element is a sequence of predefined matching
 2624 strings. The use of `xs:sequence` is required to allow more than one instance of a set of
 2625 `fillMatches`. It should not be interpreted as implying any particular order. However, the order
 2626 of the `matchText` elements within the `xs:sequence` for each `fillMatches` element is sig-
 2627 nificant. The W3C XML Schema definition language has no provision to express this difference
 2628 of interpretation of `xs:sequence`. Implementations should be aware of the data-model re-
 2629 quirements.

```

2630     <xs:group name="grpCorrectFillIn">
2631       <xs:sequence>
2632         <xs:element name="fillMatches" minOccurs="0"
2633           maxOccurs="unbounded">
2634           <xs:complexType>
2635             <xs:sequence>
2636               <xs:element name="matchText"
2637                 type="localizedString250Type" maxOccurs="unbounded"/>
2638             </xs:sequence>
2639             <xs:attribute name="caseMatters"
2640               type="trueFalseType" use="optional" default="false"/>
2641             <xs:attribute name="orderMatters" type="trueFalseType"
2642               use="optional" default="true"/>
2643             <xs:attribute name="collectionType" fixed="array"/>
2644             <xs:attribute name="spm" fixed="10"/>
2645           </xs:complexType>
2646         </xs:element>
2647       </xs:sequence>
2648     </xs:group>
  
```

2649 **D.6.2 Correct responses for likert**

2650 In this group definition, the `likert` choice is a single, optional identifier.

```

2651     <xs:group name="grpCorrectLikert">
2652       <xs:sequence>
2653         <xs:element name="choice" type="shortIdentifierType"
2654           minOccurs="0"/>
2655       </xs:sequence>
2656     </xs:group>
  
```

2657 **D.6.3 Correct responses for long fill-in**

2658 In this group definition, each `matchText` element is a single, predefined, matching string with
 2659 an optional attribute specifying whether case matters for this match. The use of `xs:sequence` is
 2660 required to allow more than one instance of a set of `matchText`. It should not be interpreted as
 2661 implying any particular order.

```

2662     <xs:group name="grpCorrectLongFillIn">
2663       <xs:sequence>
2664         <xs:element name="matchText" maxOccurs="unbounded">
2665           <xs:complexType>
2666             <xs:simpleContent>
2667               <xs:extension base="localizedString4000Type">
2668                 <xs:attribute name="caseMatters"
2669                   type="trueFalseType" use="optional" default="false"/>
2670               </xs:extension>
2671             </xs:simpleContent>
2672           </xs:complexType>
2673         </xs:element>
2674       </xs:sequence>
2675     </xs:group>
  
```

2676 **D.6.4 Correct responses for matching**

2677 In this group definition, each `matchPattern` element of type `responseMatchingType` repre-
 2678 sents a set of matching pairs. The use of `xs:sequence` is required to allow more than one in-
 2679 stance of `matchPattern`. It should not be interpreted as implying any particular order.

```

2680     <xs:group name="grpCorrectMatching">
2681       <xs:sequence>
2682         <xs:element name="matchPattern" type="matchingPairsType"
2683           minOccurs="0" maxOccurs="unbounded"/>
2684       </xs:sequence>
2685     </xs:group>
  
```

2686 The global type `matchingPairsType` implements a match pattern, which is a collection of
 2687 matching pairs. No order is implied. Pairs need not be unique.

```

2688     <xs:complexType name="matchingPairsType">
2689       <xs:sequence>
2690         <xs:element name="pair" type="matchingPairType"
2691           minOccurs="0" maxOccurs="unbounded"/>
2692       </xs:sequence>
2693     </xs:complexType>
  
```

2694 The global type `matchingPairType` implements a single matching pair, which is defined as an
 2695 empty element with `source` and `target` attributes.

```

2696     <xs:complexType name="matchingPairType">
2697       <xs:attribute name="source" type="shortIdentifierType"/>
2698       <xs:attribute name="target" type="shortIdentifierType"/>
2699     </xs:complexType>
  
```

2700 **D.6.5 Correct responses for multiple choice**

2701 This element is defined in the data model as set of sets of short identifiers, each of which repre-
2702 sents a choice. The outer set is a collection of `choices` elements. The use of `xs:sequence` is
2703 required to allow more than one instance of a set of choices. It should not be interpreted as im-
2704 plying any particular order.

```
2705     <xs:group name="grpCorrectMultipleChoice">
2706       <xs:sequence>
2707         <xs:element ref="choices" minOccurs="0"/>
2708       </xs:sequence>
2709     </xs:group>
```

2710 The global element `choices` is of type `bagOfChoiceTypes`, which is a collection of choice
2711 elements. The use of `xs:unique` is required to constrain the collection to a set of unique
2712 choice elements.

```
2713     <xs:element name="choices" type="bagOfChoicesType">
2714       <xs:unique name="uniqueInChoicesIds">
2715         <xs:selector xpath="./choice"/>
2716         <xs:field xpath="."/>
2717       </xs:unique>
2718     </xs:element>
```

2719 The type `bagOfChoiceTypes` is a collection of choice elements. The use of `xs:sequence`
2720 should not be interpreted as implying any particular order.

```
2721     <xs:complexType name="bagOfChoicesType">
2722       <xs:sequence>
2723         <xs:element name="choice" type="shortIdentifierType"
2724           minOccurs="0" maxOccurs="unbounded"/>
2725       </xs:sequence>
2726     </xs:complexType>
```

2727 **D.6.6 Correct responses for numeric**

2728 This group definition contains two optional elements that specify the min and max values for the
2729 correct response.

```
2730     <xs:group name="grpCorrectNumeric">
2731       <xs:sequence>
2732         <xs:element name="min" type="real7Type" minOccurs="0"/>
2733         <xs:element name="max" type="real7Type" minOccurs="0"/>
2734       </xs:sequence>
2735     </xs:group>
```

2736 **D.6.7 Correct responses for other**

2737 This group definition contains a generic, literal-string element.

```
2738     <xs:group name="grpCorrectOther">
2739       <xs:sequence>
2740         <xs:element name="correctOther" type="literalString4000Type"/>
2741       </xs:sequence>
2742     </xs:group>
```

2743 **D.6.8 Correct responses for performance**

2744 In this group definition, each performancePattern element of type
 2745 correctPerformancePatternType represents a predefined sequence of steps with associ-
 2746 ated data. The use of xs:sequence is required to allow more than one instance of
 2747 performancePattern. It should not be interpreted as implying any particular order.

```
2748     <xs:group name="grpCorrectPerformance">
2749       <xs:sequence>
2750         <xs:element name="performancePattern"
2751           type="correctPerformancePatternType" maxOccurs="unbounded"/>
2752       </xs:sequence>
2753     </xs:group>
```

2754 The global type correctPerformancePatternType specifies the encoding of a single per-
 2755 formance response pattern, which is a sequence of steps with an optional, additional, literal or
 2756 numeric, answer element specified for each step. Each step is identified by the value of the ele-
 2757 ment stepName. The order of steps is significant.

```
2758     <xs:complexType name="correctPerformancePatternType">
2759       <xs:sequence>
2760         <xs:element name="step" minOccurs="0" maxOccurs="unbounded">
2761           <xs:complexType>
2762             <xs:all>
2763               <xs:element name="stepName" type="shortIdentifierType"
2764                 minOccurs="0"/>
2765               <xs:element name="stepAnswer" minOccurs="0">
2766                 <xs:complexType>
2767                   <xs:choice>
2768                     <xs:element name="literal"
2769                       type="literalString250Type" minOccurs="0"/>
2770                     <xs:element name="numeric" minOccurs="0">
2771                       <xs:complexType>
2772                         <xs:attribute name="min" type="real7Type"/>
2773                         <xs:attribute name="max" type="real7Type"/>
2774                       </xs:complexType>
2775                     </xs:choice>
2776                   </xs:complexType>
2777                 </xs:element>
2778               </xs:all>
2779             </xs:complexType>
2780           </xs:element>
2781         </xs:sequence>
2782       <xs:attribute name="orderMatters" type="trueFalseType"
2783         use="optional" default="true"/>
2784       <xs:attribute name="collectionType" fixed="array"/>
2785       <xs:attribute name="spm" fixed="250"/>
2786     </xs:complexType>
```

2788 D.6.9 Correct responses for sequencing

2789 In this group definition, each `stepSequence` element of type `stepSequenceType` represents a
2790 predefined sequence of steps. The use of `xs:sequence` is required to allow more than one in-
2791 stance of `stepSequence`. It should not be interpreted as implying any particular order.

```
2792 <xs:group name="grpCorrectSequencing">
2793   <xs:sequence>
2794     <xs:element name="stepSequence" type="stepSequenceType"
2795       maxOccurs="unbounded" />
2796   </xs:sequence>
2797 </xs:group>
```

2798 The type `stepSequenceType` represents a sequence of steps. The use of `xs:sequence` should
2799 be interpreted here as implying a specific order.

```
2800 <xs:complexType name="stepSequenceType">
2801   <xs:sequence>
2802     <xs:element name="step" type="shortIdentifierType" minOccurs="0"
2803       maxOccurs="unbounded" />
2804   </xs:sequence>
2805   <xs:attribute name="collectionType" fixed="array" />
2806   <xs:attribute name="spm" fixed="36" />
2807 </xs:complexType>
```

2808 D.6.10 Correct responses for true false

2809 In this group definition, the true–false choice is a token with the value `true` or the value `false`.

```
2810 <xs:group name="grpCorrectTrueFalse">
2811   <xs:sequence>
2812     <xs:element name="trueOrFalse" type="trueFalseType" />
2813   </xs:sequence>
```

2814 The global type `trueFalseType` implements true and false options for the interaction type
2815 `true_false` specified in subclauses 6.1.9.5 and 6.1.9.7 of IEEE 1484.11.1–2004. This global
2816 type is also used for tokens for various other Boolean elements.

```
2817 <xs:simpleType name="trueFalseType">
2818   <xs:restriction base="xs:token">
2819     <xs:enumeration value="true" />
2820     <xs:enumeration value="false" />
2821   </xs:restriction>
2822 </xs:simpleType>
```

2823 D.6.11 Learner response for fill-in

2824 In this group definition, the learner response consists of a sequence of zero or more
2825 `fillString` elements of type `localizedString250Type`.

```
2826 <xs:group name="grpResponseFillIn">
2827   <xs:sequence>
2828     <xs:element name="fillString" type="localizedString250Type"
2829       minOccurs="0" maxOccurs="unbounded">
2830   </xs:element>
2831 </xs:sequence>
```

2832 </xs:group>

2833 **D.6.12 Learner response for likert**

2834 In this group definition, the learner response is a single, optional identifier.

```
2835       <xs:group name="grpResponseLikert">
2836        <xs:sequence>
2837         <xs:element name="choice" type="shortIdentifierType"
2838           minOccurs="0"/>
2839        </xs:sequence>
2840       </xs:group>
```

2841 **D.6.13 Learner response for long fill-in**

2842 In this group definition, the learner response is a single localized string.

```
2843       <xs:group name="grpResponseLongFillIn">
2844        <xs:sequence>
2845         <xs:element name="longFillString" type="localizedString4000Type"
2846           minOccurs="0"/>
2847        </xs:sequence>
2848       </xs:group>
```

2849 **D.6.14 Learner response for matching**

2850 In this group definition, the learner response is a match pattern, which is a collection of matching
2851 pairs. No order is implied. Pairs need not be unique.

```
2852       <xs:group name="grpResponseMatching">
2853        <xs:sequence>
2854         <xs:element name="matchPattern" type="matchingPairsType"/>
2855        </xs:sequence>
2856       </xs:group>
```

2857 **D.6.15 Learner response for multiple choice**

2858 In this group definition, the learner response is a list of identifiers, each of which specifies one
2859 choice. No order should be implied.

```
2860       <xs:group name="grpResponseMultipleChoice">
2861        <xs:sequence>
2862         <xs:element name="choices" type="setOfChoicesType"/>
2863        </xs:sequence>
2864       </xs:group>
2865       <xs:group name="grpResponseNumeric">
2866        <xs:sequence>
2867         <xs:element name="number" type="real7Type" minOccurs="0"/>
2868        </xs:sequence>
2869       </xs:group>
2870       <xs:group name="grpResponseOther">
2871        <xs:sequence>
2872         <xs:element name="responseOther" type="literalString4000Type"/>
2873        </xs:sequence>
2874       </xs:group>
```


2875 **D.6.16 Learner response for numeric**

2876 In this group definition, the learner response is a single numeric value.

```
2877     <xs:group name="grpResponseNumeric">
2878       <xs:sequence>
2879         <xs:element name="number" type="real7Type" minOccurs="0"/>
2880       </xs:sequence>
2881     </xs:group>
```

2882 **D.6.17 Learner response for other**

2883 In this group definition, the learner response is a single literal string.

```
2884     <xs:group name="grpResponseOther">
2885       <xs:sequence>
2886         <xs:element name="responseOther" type="literalString4000Type"
2887           minOccurs="0"/>
2888       </xs:sequence>
2889     </xs:group>
```

2890 **D.6.18 Learner response for performance**

2891 In this group definition, the learner response is a single performance pattern that represents the
2892 actual sequence of steps with associated data.

```
2893     <xs:group name="grpResponsePerformance">
2894       <xs:sequence>
2895         <xs:element name="step" type="learnerPerformanceStepType"
2896           minOccurs="0" maxOccurs="unbounded"/>
2897       </xs:sequence>
2898     </xs:group>
```

2899 The type `learnerPerformanceStepType` implements the learner response elements for a sin-
2900 gle performance step.

```
2901     <xs:complexType name="learnerPerformanceStepType">
2902       <xs:all>
2903         <xs:element name="stepName" type="shortIdentifierType"
2904           minOccurs="0"/>
2905         <xs:element name="stepAnswer" minOccurs="0">
2906           <xs:complexType>
2907             <xs:choice>
2908               <xs:element name="literal" type="literalString250Type"
2909                 minOccurs="0"/>
2910               <xs:element name="numeric" type="real7Type"
2911                 minOccurs="0"/>
2912             </xs:choice>
2913           </xs:complexType>
2914         </xs:element>
2915       </xs:all>
2916     </xs:complexType>
```

2917 D.6.19 Learner response for sequencing

2918 In this group definition, the learner response is a list of steps identifiers.

```
2919     <xs:group name="grpResponseSequencing" >
2920       <xs:sequence>
2921         <xs:element name="steps" type="stepSequenceType" minOccurs="0" />
2922       </xs:sequence>
2923     </xs:group>
```

2924 D.6.20 Learner response for true false

2925 In this group definition, the learner response is a sequence of zero or more step identifiers.

```
2926     <xs:group name="grpResponseTrueFalse" >
2927       <xs:sequence>
2928         <xs:element name="trueOrFalse" type="trueFalseType"
2929           minOccurs="0" />
2930       </xs:sequence>
2931     </xs:group>
```

2932 D.6.21 Matching response type

2933 This type implements a set of matching pairs. No particular order is implied.

```
2934     <xs:complexType name="responseMatchingType" >
2935       <xs:sequence>
2936         <xs:element name="pair" type="matchingPairType"
2937           maxOccurs="unbounded" >
2938       </xs:element>
2939     </xs:sequence>
```

2940 **Annex E**

2941 (informative)

2942 **Internet availability of the XSD file and example**
2943 **instance**

2944 The XSD file in Annex B and the example instance in Annex C are available on the World Wide
2945 Web at the following URL:

2946 – <http://standards.ieee.org/reading/ieee/downloads/1484.11.3-2005/>
2947

2948 The XSD file is available for downloading and for direct inclusion in applications.