IEEE

1484.20.1™

IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions

IEEE Computer Society

Sponsored by the
Learning Technology Standards Committee

IEEE Std 1484.20.1™-2007

IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions

Sponsor

Learning Technology Standards Committee
of the
IEEE Computer Society

Approved 27 September 2007
IEEE-SA Standards Board

This Standard is based on the IMS Reusable Definition of Competency or Educational Objective Specification, Version 1.0, published on October 25, 2002, by the IMS Global Learning Consortium, Inc.

Abstract: This Standard defines a data model for describing, referencing, and sharing competency definitions, primarily in the context of online and distributed learning. This Standard provides a way to represent formally the key characteristics of a competency, independently of its use in any particular context. It enables interoperability among learning systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

Keywords: competency, competency definition, reusable competency definition

Introduction

This introduction is not part of IEEE Std 1484.20.1-2007, IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions.

This Standard defines a data model for describing, referencing, and sharing competency definitions, primarily in the context of online and distributed learning. This Standard provides a way to represent formally the key characteristics of a competency, independently of its use in any particular context. It enables interoperability among learning systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

Dedication

The LTSC Reusable Competency Definitions Working Group dedicates this standard to the memory of Claude Ostyn. As chair of the LTSC 1484.20 Working Group, he provided initiative, guidance, and technical insight that made this standard possible. His hard work is much appreciated and will be remembered.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions are reasonable or non-discriminatory. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard submitted to the IEEE-SA Standards Board for approval, the Competency Data Standards Working Group had the following membership:

Claude Ostyn, *Chair**
Scott Lewis, *Technical Editor*

Michael Brown	Jason L. Haag	Robby Robson
Mike Collett	Chad J. Kainz	Christian Stracke
Geoffrey Darnton	Fanny Klett	Schawn E. Thropp
Geoffrey A. Frank	Rolf Lindner	Luk Vervenne
	Daniel R. Rehak	

*Deceased

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

William J. Ackerman	Werner Hoelzl	Michael S. Newman
Juan C. Carreon	Dennis Horwitz	Claude Ostyn
Keith Chow	Atushi Ito	Daniel R. Rehak
Paul Colcher	Chad J. Kainz	Robert A. Robinson
Tommy P. Cooper	Piotr Karocki	Robby Robson
Geoffrey Darnton	Robert B. Kelsey	Randall M. Safier
Thomas J. Dineen	Fanny Klett	Bartien Sayogo
Kameshwar V. Eranki	Mark J. Knight	Colin Smythe
Allan M. Gillard	Susan Land	Thomas E. Starai
John Garth Glynn	Rolf Lindner	Christian Stracke
Randall C. Groves	Gregory Luri	Schawn E. Thropp
Jason L. Haag	Jonathan C. Mason	Mark A. Tillinghast
Rutger A. Heunks	Gary Michel	Derek T. Woo
	Georges F. Montillet	

When the IEEE-SA Standards Board approved this guide on 27 September 2007, it had the following membership:

Steve M. Mills, *Chair*
Robert M. Grow, *Vice Chair*
Don Wright, *Past Chair*
Judith Gorman, *Secretary*

Richard DeBlasio	Hermann Koch	Narayanan Ramachandran
Alex Gelman	Joseph L. Koepfinger*	Greg Ratta
William R. Goldbach	John Kulick	Robby Robson
Arnold M. Greenspan	David J. Law	Anne-Marie Sahazizian
Joanna N. Guenin	Glenn Parsons	Virginia C. Sulzberger
Kenneth S. Hanus	Ronald C. Petersen	Malcolm V. Thaden
William B. Hopf	Tom A. Prevost	Richard L. Townsend
Richard H. Hulett		Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Michael H. Kelley, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions

1. Overview

1.1 Scope

This Standard defines a data model for describing, referencing, and sharing competency definitions, primarily in the context of online and distributed learning. This Standard provides a way to represent formally the key characteristics of a competency, independently of its use in any particular context. It enables interoperability among learning systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

This Standard enables information about competencies to be encoded and shared. This Standard does not define whether a competency is a skill, knowledge, ability, attitude, or learning outcome, but it can be used to capture information about any of these. Policies regarding reusable competency definitions (RCDs), such as the best practice to look for an existing definition to reuse instead of inventing a new one for the same purpose, are outside of the scope of this Standard.

1.2 Purpose

This Standard is based on the existing IMS Global Learning Consortium (IMS) specification entitled IMS Reusable Definition of Competency or Educational Objective (RDCEO) [B4].[1] Implementations that conform to the IMS specification will conform to this Standard.

---

[1] The numbers in brackets correspond to those of the bibliography in Annex A.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1484.12.1™-2002, IEEE Standard for Learning Object Metadata.[1, 2]

IETF RFC 3986:2005, Uniform Resource Identifiers (URI): Generic Syntax.[3]

ISO 639–1, Code for the representation of names of languages—Part 1: Alpha-2 code.[4]

ISO 639–2, Codes for the representation of names of languages—Part 2: Alpha-3 code.

ISO 3166–1, Codes for the representation of names of countries and their subdivisions—Part 1: Country codes.

ISO/IEC 10646, Information technology—Universal Multiple-Octet Coded Character Set (UCS).[5]

ISO/IEC 11404:1996, Information technology—Programming languages, their environments and system software interfaces—Language-independent datatypes.[6]

W3C Recommendation (28 October 2004), XML Schema Part 2: Datatypes Second Edition.[7]

3. Definitions, acronyms, and abbreviations

3.1 Definitions

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

3.1 competency definition record: An instance of a data structure that represents an identifiable competency definition.

---

[1] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[2] The IEEE standards or products referred to in this introduction are trademarks of the Institute of Electrical and Electronics Engineers, Inc.
[3] IETF publications are available from the Internet Engineering Task Force website at http://ietf.org/rfc.html.
[4] ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH–1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http:/www.ansi.org/).
[5] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH–1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http:/www.ansi.org/).
[6] ISO/IEC 11404:1996 may be downloaded at no cost from http://www.iso.org/.
[7] W3C publications are available from the World Wide Web Consortium, 32 Vassar Street, Room 32–G515, Cambridge, MA 02139, USA (http://www.w3.org/).

3.2 LangString: A data type that represents one or more character strings. A LangString value may include multiple, semantically equivalent, character strings, such as translations or expressions of a description in different languages.

3.2 Acronyms and abbreviations

IMS   IMS Global Learning Consortium
RCD   reusable competency definition
RDCEO   Reusable Definition of Competency or Educational Objective
SPM   smallest permitted maximum
URI   Uniform Resource Identifier
XML   Extensible Markup Language

4. Special terms

In this Standard, the following special terms and definitions apply.

competency: Any aspect of competence, such as knowledge, skill, attitude, ability, or learning objective.

NOTE—In this Standard, the term "competency" is to be interpreted in the broadest sense to include learning objectives (those things that are sought) as well as competencies (those things that are achieved). The term "competency" is also used to include all classes of things that someone, or potentially something, can be competent in, although some communities of practice use the term with nuance, for example, limiting its use to skill and excluding knowledge or understanding.[8]

5. Conformance

5.1 Shall and shall not

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

5.2 RCD instances

A conforming RCD instance shall conform to the data model as specified in Clause 5.

5.3 Smallest permitted maximum values

This standard defines smallest permitted maximum (SPM) values for many data elements. The SPM is the number of entries that a conforming implementation is required to process for a given data element in an RCD instance. An RCD instance may contain more entries for a data element than the SPM. An implementation that receives entries in excess of the SPM shall not produce an error. The implementation may, but is not required to, process those entries. For the purposes of the SPM, an entry may be an element within a collection, such as a bag or record, or it may be a character in a string. SPM values are defined for

---

[8] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the guide.

— **Items in a collection:** For example, an SPM is defined for Statement elements within a Definition. If the SPM for the number of Statement elements is n but the number of statements in an RCD instance is $n + 1$, an implementation that receives the Statement elements shall not produce an error. The implementation shall process at least n elements but may ignore the element that exceeds the SPM.

— **Character strings defined in the types CharacterString and LangString:** For example, if a string found in an element of one of these types in an RCD instance contains more characters than specified by the SPM, an implementation that receives the string shall not produce an error but may ignore the extra characters.

NOTE 1—The intent is for the SPM values to cover most cases.

NOTE 2—The meaning of "process" in this subclause depends on the nature of the application.

NOTE 3—This Standard does not define any provision for whether or how a system may process more than the SPM value for a particular data element.

NOTE 4—This Standard does not define failure conditions if a system cannot process the SPM value for a particular data element.

NOTE 5—If the SPM for the number of elements in a collection is n, but the number of elements is $n + 1$, the selection of which n elements an implementation is required to process is left to the implementation.

6. Data model

6.1 General information

Clause 6 defines the data elements of an RCD. Unless otherwise noted, all components of records are optional in an RCD instance. The data elements of an RCD are described in 6.2. The auxiliary data types used in conjunction with the data elements are described in 6.3. (See Annex B for a conceptual overview of RCDs and the data model.)

NOTE 1—The use of ISO/IEC 11404:1996[9] notation in the synopses in 6.2 and 6.3 is for informative purposes only and is not meant to prescribe a specific implementation. A complete implementation of the operations defined in ISO/IEC 11404:1996 is not required for conformance.

NOTE 2—The ISO/IEC 11404:1996 notation describes the semantics of the language-independent data types across all bindings (e.g., implementation of a data type as itself, its subtypes, its subclasses, and its specializations). For example, an ISO/IEC 11404:1996 record may be implemented as an Extensible Markup Language (XML) complexType; an ISO/IEC 11404:1996 characterstring may be implemented in an encoding (ISO/IEC 646 [B5], ISO/IEC 8859-1 [B6], UTF-8 [B2], UTF-16 [B3], UTF-32 [B7], etc.) that supports the repertoire specified for the characterstring data type.

NOTE 3—All examples in 6.2 and 6.3 are informative and do not endorse any particular binding.

NOTE 4—The following language-independent data types used in this Standard are defined in the ISO/IEC 11404:1996: bag, characterstring, and record.

NOTE 5—The labels for data elements and data types in the synopses in 6.2 and 6.3 are for reference only. An implementation is not required to use the same labels, as long as the data elements and data types are semantically equivalent.

---

[9] For information on normative references, see Clause 2.

NOTE 6—This Standard does not define a specific extension mechanism for the data model. Implementers may define bindings that allow additional elements or create additional data models for competency data. Such models may be used to augment this model to support different application domains. Extensions should not duplicate or redefine the semantics of elements defined in this Standard.

NOTE 7—The inclusion of extension elements in an RCD instance may cause interoperability issues with some implementations or bindings. Therefore, extensions should be provided by including the RCD data model inside an extended data model, rather than by inserting extension elements within the RCD data model itself.

NOTE 8—The RCD data model can be used in profiles and data models defined in particular implementations or in other standards. For example, a profile for learning objectives or assessable competencies might not define extension elements, but it might extend the conformance requirements to specify that a Definition element is mandatory in every RCD instance that conforms to the profile and that the Definition element has to contain several Statement elements with specific names. Another implementation might define a data model for a collection of RCD instances. Yet another implementation might define a map or graph to represent a competency model, with the nodes in the graph referencing RCD instances through their unique identifier.

6.2 Reusable competency definition

Synopsis

```
reusable_competency_definition :
record
(
   identifier :
      long_identifier_type,
         // Mandatory
         // Occurs 1 time
   title :
      bag of langstring_type(1000),
         // Mandatory
         // Occurs 1 time
         // SPM: 20 instances of langstring_type in the bag
         // The parameter value is the SPM for the number of
         // of characters in the string element of the
         // langstring_type
   description :
      bag of langstring_type(4000),
         // Optional
         // Occurs 0 or 1 times
         // SPM: 20 instances of langstring_type in the bag
         // The parameter value is the SPM for the number of
         // of characters in the string element of the
         // langstring_type
   definition :
      definition_type,
         // Optional
         // Occurs 0 or more times
         // SPM: 10 instances of definition in a
         // reusable_competency_definition record
```

```
          metadata :
             metadata_type,
                // Optional (implied default values - see 6.2.5)
                // Occurs 0 or 1 times
)
```

Definition

The components of `reusable_competency_definition` are defined in 6.2.1 through 6.2.5. Identifier and Title are mandatory and shall be included in RCD instances. Depending on the implementation, an instance of `reusable_competency_definition` may include any or all of the elements Description, Definition, and Metadata.

All elements in this data model are intrinsically unordered. The order of the elements in the data model synopses and the order of the values in a list of values are not significant. For example, if an RCD instance includes three Definition elements, their order is not significant. They may appear in any order without loss of information.

NOTE—A binding may impose a particular ordering on RCD instances that conform to that binding.

6.2.1 Identifier

Synopsis

```
   identifier :
      long_identifier_type,
         // Mandatory
         // Occurs 1 time
```

Description

This mandatory data element is a globally unique label that identifies the RCD. The label shall be unique across all systems that store or process the RCD.

`long_identifier_type` is defined in 6.3.3.

NOTE 1—This data element uses the same subelements as the Identifier element defined in IEEE Std 1484.12.1-2002 and consists of two subelements, Catalog and Entry (see 6.3.3).

NOTE 2—This Standard does not specify any method to define or test the global uniqueness of the identifier.

6.2.2 Title

Synopsis

```
   title :
      bag of langstring_type(1000),
         // Mandatory
         // Occurs 1 time
         // SPM: 20 instances of langstring_type in the bag
         // The parameter value is the SPM for the number of
```

```
        // of characters in the string element of the
        // langstring_type
```

Description

This mandatory data element is a single, text label for the RCD. The label is a human-readable name for the RCD. Because different application domains may coincidentally define the same title for different competency definitions, the Identifier element (see 6.2.1), not the Title, shall be used to distinguish among RCDs. The Title element is intended to provide a title, only, and should not be used to provide descriptive detail in lieu of the Description element (see 6.2.3).

`langstring_type` is defined in 6.3.2.

NOTE 1—This data element may be expressed in multiple languages. Each translation is represented by an instantiation of `langstring_type`.

NOTE 2—Although the Identifier element (see 6.2.1) provides the definitive reference to the definition, it is typically unintelligible. The Title element provides a convenient, alternative, readable form. Examples: "English proficiency", "Schmiblick failure diagnostic level 4", and "Conflict resolution skills".

6.2.3 Description

Synopsis

```
description :
   bag of langstring_type(4000),
      // Optional
      // Occurs 0 or 1 times
      // SPM: 20 instances of langstring_type in the bag
      // The parameter value is the SPM for the number of
      // of characters in the string element of the
      // langstring_type
```

Description

This optional data element is a human-readable description of the competency. It is an unstructured character string meant to be interpretable only by humans.

`langstring_type` is defined in 6.3.2.

NOTE 1—This data element may be expressed in multiple languages. Each translation is represented by an instantiation of `langstring_type`.

NOTE 2—This data element is typically more explicative than the Title element (see 6.2.2). Examples: "Proficiency in written and spoken English and use of English for meaningful oral or written expression" and "Performance of level 4 diagnostic as specified in IETM #SCMBLK007".

6.2.4 Definition

Synopsis

```
definition :
   definition_type,
      // Optional
      // Occurs 0 or more times
      // SPM: 10 instances of definition in a
      // reusable_competency_definition record


definition_type = record
(
   model_source :
      characterstring(iso-10646),
         // Optional
         // Occurs 0 or 1 times
         // SPM: 1000 characters
   statement :
      statement_type,
         // Mandatory
         // Occurs 1 or more times
         // SPM: 10 instances of statement in a
         // definition_type record
),
```

Description

This optional data element is a structured description that provides a formal definition of the competency, usually using attributes taken from a specific model of how a competency should be structured or defined. This data element shall contain zero or one Model source elements and at least one Statement element. It may contain multiple Statement elements.

NOTE 1—Typically, the models that underlie this data element define competencies in terms of "statement, conditions, criteria," "proficiency, criteria, indicators," "standards, performance indicators, outcomes," "abilities, basic skills, content, process," and similar sets of statements.

NOTE 2—This data element provides a structure for including an arbitrary collection of statements that describe a competency. The author of an RCD is free to use this data element in any way that best describes the competency.

6.2.4.1 Model source

Synopsis

```
model_source :
   characterstring(iso-10646),
      // Optional
      // Occurs 0 or 1 times
      // SPM: 1000 characters
```

Description

This optional data element is the source of the model used for the competency definition. The characters in the string shall belong to the repertoire of ISO/IEC 10646 as allowed by IETF RFC 3986:2005.

NOTE—The value of this data element should be sufficiently specific to avoid conflict with other source names; therefore, the value should be a Uniform Resource Identifier (URI). If the value of this data element is a URI, it may point to an actual document that defines the source formally. Examples: "http://example.org/3-part-learning-objective" and "http://example.edu/ref/los.xml".

6.2.4.2 Statement

Synopsis

```
statement :
   statement_type,
      // Mandatory
      // Occurs 1 or more times

statement_type = record
(
   statement_id :
      long_identifier_type,
         // Conditionally mandatory (see description)
         // Occurs 0 or 1 times
   statement_name :
      characterstring(iso-10646),
         // Conditionally mandatory (see description)
         // Occurs 0 or 1 times
         // SPM: 1000 characters
   statement_text :
      bag of langstring_type(1000),
         // Conditionally mandatory (see description)
         // Occurs 0 or 1 times
         // SPM: 20 instances of langstring_type in the bag
         // The parameter value is the SPM for the number of
         // of characters in the string element of the
         // langstring_type
   statement_token :
      vocabulary_type,
         // Conditionally mandatory (see description)
         // Occurs 0 or 1 times
),
```

Description

Each instance of this data element is a description of a single characteristic of a Definition element (see 6.2.4). At least one Statement element is mandatory if a Definition element exists. A record of type `statement_type` shall contain one or more of the elements defined in `statement_type`. None of these elements is specifically required, but at least one shall be present.

For example, a particular learning-objective model might require a list of specific statement strings, each of which has a specific name, such as "Condition", "Performance", and "Standard". A Statement element matching this model would use the components Statement name and Statement text (see 6.2.4.2.2 and 6.2.4.2.3).

6.2.4.2.1 Statement ID

Synopsis

```
statement_id :
    long_identifier_type,
        // Conditionally mandatory (see 6.2.4.2)
        // Occurs 0 or 1 times
```

Description

This conditionally mandatory data element is a label for the statement. This label shall be unique at least within the scope of the definition.

`long_identifier_type` is defined in 6.3.3.

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

6.2.4.2.2 Statement name

Synopsis

```
statement_name :
    characterstring(iso-10646),
      // Conditionally mandatory (see 6.2.4.2)
      // Occurs 0 or 1 times
      // SPM: 1000 characters
```

Description

This conditionally mandatory data element is a name for the statement. This name shall be unique at least within the scope of the definition. Examples: "Condition", "Action", "Standard", "Outcome", and "Criteria". The characters in the string shall belong to the repertoire of ISO/IEC 10646.

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

6.2.4.2.3 Statement text

Synopsis

```
statement_text :
    bag of langstring_type(1000),
        // Conditionally mandatory (see 6.2.4.2)
        // Occurs 0 or 1 times
        // SPM: 20 instances of langstring_type in the bag
```

```
                   // The parameter value is the SPM for the number of
                   // of characters in the string element of the
                   // langstring_type
```

Description

This conditionally mandatory data element is an unstructured, textual description of those aspects of the RCD referred to by the Statement name element (see 6.2.4.2.2). Example: "Given a set of integer numbers in the range 1 to 49".

NOTE—This data element may be expressed in multiple languages. Each translation is represented by an instantiation of `langstring_type`.

6.2.4.2.4 Statement token

Synopsis

```
statement_token :
   vocabulary_type,
      // Conditionally mandatory (see 6.2.4.2)
      // Occurs 0 or 1 times

vocabulary_type = record
(
   source :
      characterstring(iso-10646),
         // Mandatory
         // Occurs 1 time
         // SPM: 1000 characters
   value :
      characterstring(iso-10646),
         // Mandatory
         // Occurs 1 time
         // SPM: 1000 characters
),
```

Description

This conditionally mandatory data element consists of a vocabulary token, along with an identifier of the token's source, which allows the use of controlled terms (vocabularies) instead of, or along with, free-form statement text (see 6.2.4.2.3).

A record of type `vocabulary_type` shall contain both a Source element and a Value element. The Source element indicates the source of the token value. The Source element may be a URI that identifies a formal vocabulary definition. Example: "http://www.example.org/OSList". The Value element is the actual token value from a list of tokens defined in the source. For example, the token might be `MRS_15`.

NOTE 1—This approach to controlled terms (vocabularies) follows that used in metadata standards such as IEEE Std 1484.12.1-2002. In this Standard, the vocabulary token is just a string value associated with the specification of a source. It is assumed that the source defines a context in which the value is meaningful. The value does not have to be a human-language word and does not have to be meaningful out of context. The source typically defines the meaning of the token, either by reference to a standard or by the fact that the data in the source element is a URI for a human- or machine-readable description of the vocabulary tokens.

NOTE 2—This Standard does not define what a source is, only that the source has an identifier. For example, a source may be another standard, a policy document, or a formal vocabulary.

NOTE 3—This Standard does not specify how vocabularies are created, assigned, or resolved.

6.2.5 Metadata

Synopsis

```
metadata :
   metadata_type,
   // Optional
   // Occurs 0 or one times

metadata_type = record
(
   rcd_schema :
      characterstring(iso-10646),
         // Optional, defined default value
         // Occurs 0 or 1 times
         // SPM: 1000 characters
   rcd_schema_version :
      characterstring(iso-10646),
         // Optional, defined default value
         // Occurs 0 or 1 times
         // SPM: 1000 characters
   additional_metadata :
      any_type,
         // Optional
         // Occurs 0 or more times
         // SPM: 10 of instances of additional_metadata
         // in a metadata_type record
),
```

Description

This optional data element consists of embedded metadata about the RCD. This data element does not preclude the use of external metadata about the RCD. Such external metadata are not defined by this Standard.

`any_type` is defined in 6.3.1.

NOTE—Application profiles may specify additional metadata requirements.

6.2.5.1 RCD schema

Synopsis

```
rcd_schema :
   characterstring(iso-10646),
      // Optional, defined default value
```

```
// Occurs 0 or 1 times
// SPM: 1000 characters
```

Description

This optional data element is a label for the schema that defines and controls the RCD instance. The characters in the string shall belong to the repertoire of ISO/IEC 10646. If this data element is omitted from the RCD instance, then a value of "ieee.org/1484.20.1/2007" shall be assumed.

NOTE 1—Different values may be used to indicate that RCD instances conform to particular application profiles but should not be used to replicate the purpose of other elements such as Model source (see 6.2.4.1).

NOTE 2—This data element is not a label for the schema of the embedded metadata defined in 6.2.5.3. Every instance of embedded metadata, if any, should include its own schema description or identifier.

6.2.5.2 RCD schema version

Synopsis

```
rcd_schema_version :
    characterstring(iso-10646),
        // Optional
        // Occurs 0 or 1 times
        // SPM: 1000 characters
```

Description

This optional data element describes the version of the schema described by the RCD schema element (see 6.2.5.1). The characters in the string shall belong to the repertoire of ISO/IEC 10646. If this element is omitted, then a value of "1.0" shall be assumed.

6.2.5.3 Additional metadata

Synopsis

```
additional_metadata :
    any_type,
        // Optional
        // Occurs 0 or more times
```

Description

Zero or more instances of additional, embedded metadata that describe the RCD instance may be included in the metadata element. If additional metadata are present, the actual type of each element shall be defined by an application profile. An implementation shall accept any additional metadata record, but it is not required to interpret such additional metadata records.

`any_type` is defined in 6.3.1.

NOTE 1—If only one additional metadata record is included, the record should conform to IEEE Std 1484.12.1-2002. Useful metadata defined in IEEE Std 1484.12.1-2002 include additional identification as an entry in one or more catalogs, information about the author, the creation date, and so on. The IEEE 1484.12.1 Relation element may be used to relate a competency definition to a prior version of the definition. One or more IEEE 1484.12.1 Classification elements may be used to indicate where this particular definition fits in a taxonomy of competencies.

NOTE 2—A particular binding specification or application profile may impose additional restrictions or requirements.

NOTE 3—Each additional metadata record should contain information about the name and version of its schema. For example, in IEEE 1484.12.1 metadata records, this is specified in the meta-metadata element of the metadata record.

6.3 Auxiliary data types

The data types described in 6.3.1 through 6.3.3 are used in conjunction with the data elements described in 6.2.

6.3.1 Any tape

Synopsis

```
type any_type = (unspecified);
```

Description

This data type represents any type not specified in this Standard. This Standard does not require an implementation to process data elements of this type when encountered in an RCD instance.

NOTE—If implementations specify or require data elements for which the type is defined in this Standard as any_type, the implementations should provide the means to interpret and validate the implementation-specific data. For example, an implementation that uses an XML binding should include a valid XML schema that can be referenced in RCD instances bound in XML documents, and the schema should be documented.

6.3.2 LangString type

Synopsis

```
type langstring_type(length) = record
(
   language :
      characterstring(iso-10646),
         // Mandatory
         // Occurs 1 time
         // SPM: 250 characters
   string :
      characterstring(iso-10646),
         // Mandatory
         // Occurs 1 time
         // SPM: the length parameter
);
```

Description

This data type consists of a language specification for a localized string and the string itself.

Examples

Below are two examples of localized strings: `"Information Technology"` in French and `"localization"` in British English:

```
"fr", "Technologies de l'information"
"en-GB", "localisation"
```

6.3.2.1 Language

Synopsis

```
language :
   characterstring(iso-10646),
      // Mandatory
      // Occurs 1 time
      // SPM: 250 characters
```

Description

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

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

— Two-letter codes are defined by ISO 639–1.

— Three-letter codes are defined by ISO 639–2.

— The value prefix `"i"` is reserved for registrations defined by the Internet Assigned Numbers Authority (IANA).

— The value prefix `"x"` is reserved for private use.

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

— Two-letter subcodes are ISO 3166–1 alpha-2 country codes.

— Subcodes of from three to eight letters are registered with IANA.

Constraints for additional subcodes are unspecified.

The character string shall conform to the syntax defined for a valid language code by the XML Schema derived data type language (see W3C Recommendation XML Schema, Part 2 [28 October 2004]).

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

NOTE 1—The language code is normally given in lowercase and the subcodes (if any) in uppercase. However, the values are case insensitive.

NOTE 2—The syntax defined by the XML Schema derived data type `language` does not enforce all constraints on this language code.

Examples

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

6.3.2.2 String

Synopsis

```
string :
   characterstring(iso-10646),
      // Mandatory
      // Occurs 1 time
      // SPM: The length parameter
```

Description

This data element contains the text of the localized string. The characters in the string shall belong to the repertoire of ISO/IEC 10646.

6.3.3 Long identifier type

Synopsis

```
type long_identifier_type = record
(
   catalog: characterstring(iso-10646),
      // Mandatory
      // Occurs 1 time
      // SPM: 4000 characters
   entry: characterstring(iso-10646),
      // Mandatory
      // Occurs 1 time
      // SPM: 4000 characters
);
```

Description

This data type is an identifier (a label) that is intended to be unique among all contexts in which the identifier may be used. The Catalog element is the name or designator of the identification or cataloging scheme for this entry, in other words, a namespace-scheme. The Entry element is the value of the identifier

within the identification or cataloging scheme that designates or identifies this RCD, in other words, a namespace-specific string. The characters in the strings shall belong to the repertoire of ISO/IEC 10646.

Values for `long_identifier_type` as a whole shall conform to the syntax for URIs as defined by IETF RFC 3986:2005. The catalog and entry values may be concatenated as a single character string in an application profile or binding. If the catalog and entry values are concatenated, the resulting character string shall be a globally unique identifier in the form of a URI. The identifier shall be unique across all systems that store or process the RCD. The total length of a concatenated identifier shall be no longer than 4000 characters.

NOTE—This Standard does not specify how to constrain the length of the Catalog and Entry elements in order to limit the concatenated identifier to 4000 characters. A binding may require one or more characters for concatenation of the strings contained in catalog and entry. Concatenation into a URI may require at least seven additional characters. RCD instances should allow for the possibility that the Catalog and Entry elements may be split and concatenated if the RCD record is transformed for use in a different binding. Therefore, in practice, the total length of the Catalog and Entry elements, exclusive of concatenation characters, should be constrained to at most 3990 characters.

Annex A

(informative)

Bibliography

[B1] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition, New York, Institute of Electrical and Electronics Engineers, Inc.[10, 11]

[B2] IETF RFC 3629:2003, UTF-8, a transformation format of ISO 10646.[12]

[B3] IETF RFC 2781:2000, UTF-16, an encoding of ISO 10646.

[B4] IMS Specification (25 October 2002), IMS Reusable Definition of Competency or Educational Objective, Version 1.0.[13]

[B5] ISO/IEC 646, Information technology —ISO 7-bit coded character set for information interchange.[14]

[B6] ISO/IEC 8859–1, Information processing—8-bit single-byte coded graphic character sets. Part 1—Latin alphabet No. 1.

[B7] The Unicode Consortium, *The Unicode Standard, Version 4.0*. Boston, MA: Addison-Wesley Developers Press, 2003.[15]

---

[10] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards/ieee.org/).
[11] The IEEE standards or products referred to in this introduction are trademarks of the Institute of Electrical and Electronics Engineers, Inc.
[12] IETF publications are available from the Internet Engineering Task Force website at http://ietf.org/rfc.html.
[13] IMS publications are available from the IMS Global Learning Consortium website at http://www.imsglobal.org/specifications.html.
[14] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH–1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).
[15] *The Unicode Standard, Version 4.0*, describes UTF-32.

Annex B

(informative)

Conceptual overview

B.1 Objectives

This Standard is intended to satisfy the following objectives:

— Provide a data model for competency definition records that can be shared or reused among compatible systems.

— Reconcile various existing and emerging data models into a widely acceptable model.

— Provide a way to identify the type and precision of a competency definition.

— Provide a unique identifier as the means to reference unambiguously an RCD regardless of the application in which the competency definition is stored, found, retrieved, or used. For example, metadata that describe learning content may contain references to one or more competency definition records that describe learning objectives for the content.

— Provide a data model for additional information about a competency definition, such as a title, description, and source, that is compatible with other emerging metadata standards.

— Recommend metadata as one of the methods that may be used to express how competency definitions are semantically related.

This Standard also addresses the following needs:

— A common data model that allows the building of various competency models, hierarchies, and maps. However, the definitions of such competency models, hierarchies, and maps are outside of the scope of this Standard.

— A standard method that allows persistent, long-lived competency definitions to be created, shared among systems, and maintained.

— A standard method by which competency definitions can be uniquely identified among conforming systems and repositories.

— A common data model for the metadata that give an RCD its value in a reuse environment, such as the source of the competency definition, validation information, and other meta information useful to locate an RCD in a repository or collection.

B.2 Functional overview

This Standard defines a data model for describing, referencing, and sharing competency definitions, primarily in the context of online and distributed learning. The data model provides a formal representation of the key characteristics of a competency, independently of its use in any particular context. It enables interoperability among systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

The core information in an RCD is an unstructured textual definition of the competency that can be referenced through a globally unique identifier. This definition may be refined using a user-defined model of the structure of a competency.

This Standard provides a means to capture common understandings of competencies that appear as part of learning and career plans, as learning prerequisites, or as learning outcomes. The data model in this Standard can be used to share these definitions among learning systems, human resource systems, learning content, competency or skills repositories, and other relevant systems. This Standard provides unique references to competency definitions for inclusion in other data models, such as personal competency profiles.

RCD instances that conform to this Standard are intended for interchange by machines, but most of the information they contain is intended for human interpretation.

This Standard does not address the aggregation of smaller competencies into larger competencies (e.g., "throws" plus "catches" equals "plays ball") nor does it address how competencies are to be assessed, validated, certified, recorded, or used as part of a process, such as instructional design or knowledge management. It also does not specify how records of competencies associated with an individual are structured, stored, or shared. Figure B.1 shows how an RCD integrates with competency data.
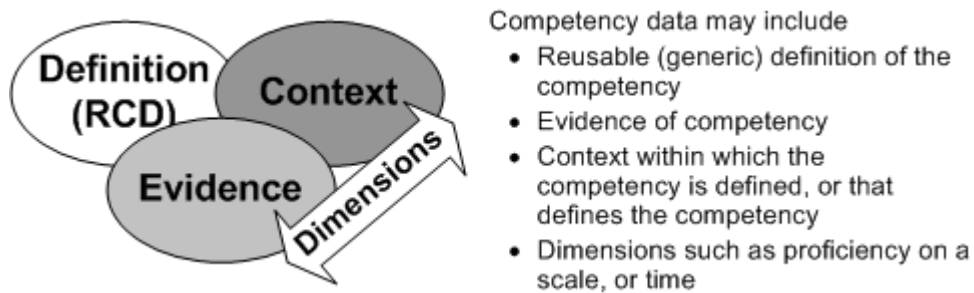

Figure B.1—RCD integration with competency data

B.3 Data model overview

The data model is minimal and extensible. It is neutral with regard to models of and uses of competencies. Competencies are defined and structured in many ways in different application domains. This Standard allows implementers to share information according to the models they use. Semantics can be "tightened" or "loosened" in the data itself, while conserving the same data model regardless of how strictly a particular organization or institution requires the data to be formulated.

The data model contains the following mandatory elements:

— **Identifier:** A globally unique label that identifies the RCD. This identifier uses the same data elements as the identifier element defined in IEEE Std 1484.12.1-2002 and consists of two subelements, Catalog and Entry. The identifier is sufficient to reference the competency in any system that stores or process the RCD.

— **Title:** A text label for the RCD. This label is a human-readable name. Although the identifier provides the definitive reference to the RCD, it is typically unintelligible. The title provides a convenient, readable alternative. The title may be repeated in multiple languages.

The data model contains the following optional elements:

— **Description:** A human-readable description of the RCD. This unstructured character string is meant to be interpretable by humans only. The description may be repeated in multiple languages.

— **Definition:** A structured description that provides a more complete definition of the RCD than the free-form description expressed in the title and description, usually using attributes taken from a specific model of how a competency should be structured or defined. Typically, such models define a competency in terms of "statement, conditions, criteria," "proficiency, criteria, indicators," "standards, performance indicators, outcomes," "abilities, basic skills, content, process," and similar sets of statements.

— **Metadata:** Embedded metadata that also describe the RCD by, for example, identifying the author and publication date of the RCD or by specifying known relations to other RCDs.

Extensibility can be achieved by defining a specific model structure within the Definition element (6.2.4) or by including elements defined by Clause 6 of IEEE Std 1484.12.1-2002 in the Metadata element (6.2.5). In addition, implementers may create additional data models that include or reference the RCD data model or RCD instances. Such implementation-specific data models may be used to augment the data model in this Standard to support different applications in various application domains.

NOTE—The identifier, not the title, is used to distinguish between RCDs, because different RCDs may coincidentally use the same title while referring to different competencies.

B.4 Taxonomies of reusable competency definitions

This Standard is intended to meet the simple need of referencing and cataloging a competency, not classifying it. Nonetheless, an implementation might want to include relation and classification information, which can be done by embedding additional metadata as specified in 6.2.5.3. Instances of RCDs also can be referenced by the nodes in a tree or by other structures representing a taxonomy or ontology of competencies.

Annex C

(informative)

An example XML binding schema

This Standard does not define any specific binding for the data model. However, related standards may reference this Standard and define bindings.

The example in Figure C.1 illustrates existing practice using an XML schema defined by the IMS Global Learning Consortium [B4].

NOTE 1—The example schema uses "rdceo" as the label for the root data element. It is not required that an implementation of this Standard use the same labels for data-element or type labels as those used in Figure C.1 as long as the elements and types themselves are semantically equivalent.

NOTE 2—The example schema conforms to this Standard. However, it allows extensions in a form different from that recommended by this Standard.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.imsglobal.org/xsd/imsrdceo_rootv1p0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.imsglobal.org/xsd/imsrdceo_rootv1p0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:group name="extelement">
    <xs:annotation>
      <xs:documentation>extension mechanism for elements</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="##other" processContents="strict"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:element name="rdceo">
    <xs:annotation>
      <xs:documentation>A single definition of a competence, educational
objective etc</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="identifier" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="title"/>
        <xs:element ref="description" minOccurs="0"/>
        <xs:element ref="definition" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="metadata" minOccurs="0"/>
        <xs:sequence minOccurs="0">
          <xs:group ref="extelement"/>
        </xs:sequence>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="strict"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="langstring">
    <xs:annotation>
```

```
    <xs:documentation>A string in a human language</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:simpleContent>
    <xs:extension base="xs:string">
     <xs:anyAttribute namespace="##other" processContents="strict"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
 </xs:element>
 <xs:element name="title">
  <xs:annotation>
   <xs:documentation>A title for the definition</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="langstring" maxOccurs="unbounded"/>
    <xs:sequence minOccurs="0">
     <xs:group ref="extelement"/>
    </xs:sequence>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="strict"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="identifier">
  <xs:annotation>
   <xs:documentation>Catenated form of the identifier of a
RDCEO</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:simpleContent>
    <xs:extension base="xs:anyURI">
     <xs:anyAttribute namespace="##other" processContents="strict"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
 </xs:element>
 <xs:element name="description">
  <xs:annotation>
   <xs:documentation>A description for the definition</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="langstring" maxOccurs="unbounded"/>
    <xs:sequence minOccurs="0">
     <xs:group ref="extelement"/>
    </xs:sequence>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="strict"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="definition">
  <xs:annotation>
   <xs:documentation>A structured form of the definition</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="model" minOccurs="0"/>
    <xs:element ref="statement" maxOccurs="unbounded"/>
    <xs:sequence minOccurs="0">
     <xs:group ref="extelement"/>
    </xs:sequence>
   </xs:sequence>
```

```xml
        <xs:anyAttribute namespace="##other" processContents="strict"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="model">
    <xs:annotation>
      <xs:documentation>The model identification for the structured
definition</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:anyAttribute namespace="##other" processContents="strict"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="statement">
    <xs:annotation>
      <xs:documentation>A component part of a structured
definition</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element ref="statementtext"/>
          <xs:element ref="statementtoken"/>
        </xs:choice>
        <xs:sequence minOccurs="0">
          <xs:group ref="extelement"/>
        </xs:sequence>
      </xs:sequence>
      <xs:attribute name="statementid" type="xs:ID"/>
      <xs:attribute name="statementname" type="xs:string"/>
      <xs:anyAttribute namespace="##other" processContents="strict"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="statementtext">
    <xs:annotation>
      <xs:documentation>Used for statements with free-form
text</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="langstring" maxOccurs="unbounded"/>
        <xs:sequence minOccurs="0">
          <xs:group ref="extelement"/>
        </xs:sequence>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="source">
    <xs:annotation>
      <xs:documentation>Source identification for a vocabulary
token</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:anyAttribute namespace="##other" processContents="strict"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
```

```
  <xs:element name="value">
   <xs:annotation>
    <xs:documentation>A vocabulary token</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:simpleContent>
     <xs:extension base="xs:string">
      <xs:anyAttribute namespace="##other" processContents="strict"/>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="metadata">
   <xs:annotation>
    <xs:documentation>A container for metadta</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="rdceoschema" minOccurs="0"/>
     <xs:element ref="rdceoschemaversion" minOccurs="0"/>
     <xs:sequence minOccurs="0">
      <xs:group ref="extelement"/>
     </xs:sequence>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="strict"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="statementtoken">
   <xs:annotation>
    <xs:documentation>Used for statements with token values (vocabulary
use)</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="source"/>
     <xs:element ref="value"/>
     <xs:sequence minOccurs="0">
      <xs:group ref="extelement"/>
     </xs:sequence>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="strict"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="rdceoschema">
   <xs:annotation>
    <xs:documentation>The identity of the RDCEO schema - assumed to be IMS
RDCEO if absent.</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:simpleContent>
     <xs:extension base="xs:string">
      <xs:anyAttribute namespace="##other" processContents="strict"/>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="rdceoschemaversion">
   <xs:annotation>
    <xs:documentation>The version of the RDCEO schema - assumed to be 1.0 if
absent</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:simpleContent>
```

```
      <xs:extension base="xs:string">
        <xs:anyAttribute namespace="##other" processContents="strict"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figure C.1—An example schema