Draft Standard for Learning Technology—Reusable Competency Definitions

THIS IS A FIRST ROUGH DRAFT, IN PROGRESS
– FOR EDITING PREVIEW ONLY

Sponsored by the Learning Technology Standards Committee
of the IEEE Computer Society

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

Abstract:

This Standard defines an information model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning. In this Standard, the word competency is used in a very general sense that includes skills, knowledge, tasks, and learning outcomes. This Standard gives a way to formally represent the key characteristics of a competency, independent 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, objective

[NOTE: Information about IEEE LTSC P1484.11 can be found at:

http://ieee.ltsc.org/wg11

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

Introduction

(This introduction is not a part of P1484.20, Draft Standard for Learning Technology—Data Model for Reusable Competency Definitions.)

This standard defines a data model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning.

Participants

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

Claude Ostyn, *Chair* Valerie Archambeau, *Technical Editor*		
Dan Rehak		
(other names to be updated)		

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

Acknowledgements

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.

Contents

Draft Standard for Learning Technology— Reusable Competency Definitions

1. Overview

The standard for Reusable Competency Definitions (RCD) defines an information model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning. The RCD standard provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning prerequisites, or as learning outcomes. RCD provides unique references to descriptions of competencies or objectives for inclusion in other information models.

"Competency" is used in a very general sense to include skills, knowledge, tasks, and learning outcomes. This standard provides a way to formally represent the key characteristics of a competency, independent 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. The information model in this standard can be used to exchange these definitions between learning systems, human resource systems, learning content, competency or skills repositories, and other relevant systems (The RCD that conform to this standard are intended for interchange by machines, but the information they contain is currently intended for human interpretation).

1.1 Scope

This standard shall specify the mandatory and optional data elements that constitute a Competency Definition as used in a Learning Management System, or referenced in a Competency Profile. This standard is intended to satisfy the following objectives:

— Provide a standardized data model for reusable Competency Definition records that can be exchanged or reused in one or more compatible systems

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

— Provide a standardized way to identify the type and precision of a Competency Definition

— Provide a unique identifier as the means to unambiguously reference are usable Competency Definition regardless of the setting in which this Competency Definition is stored, found, retrieved, or used. For example, metadata that describe learning content may contain a

reference to one or more Competency Definition records that describe the learning objectives for the content.

— Provide a standardized data model for additional information about a Competency Definition, such as a title, description, and source, compatible with other emerging learning asset metadata standards

— Provide a controlled vocabulary to express how competency definitions are semantically related.

This standard specifically does not cover:

— A data format, bindings or coding, except as minimally required for the purpose of exchange between compliant implementations

— Quality and accuracy in the data itself, although it will describe recommended best practices. For example, this standard does not cover the quality or validation of the various parts of a learning objective statement.

— A competency model, or a taxonomy of competencies.

— How the relationships between competencies are stored in a database or learning management system.

— Certification data models. However, Certification records can reference Competency Definitions. For example, an accredited authority may grant certificates that acknowledge that an individual meets the requirements for a particular competency.

— Individual competency records, as would be found in the competency profiles of individuals or groups. However, such records can include references to specific Competency Definitions. For example, a competency profile for an individual may include a collection of certificates which in turn reference Competency Definitions, as well as a collection of references to the definitions for competencies to be acquired.

1.2 Purpose

The purpose of this standard is to define a universally acceptable Competency Definition model to allow the creation, exchange and reuse of Competency Definition in applications such as Learning Management Systems, Competency or Skill Gap Analysis, Learner and other Competency profiles, etc. The standard is needed because there are currently many definitions of the terms "Learning Objective", "Competency" and "Skill", and very little agreement between how those definitions can be used to define reusable data models. This standard uses a general definition that can be semantically "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. 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 for such applications are outside the scope of this standard).

— A standard that allows persistent, long lived Competency Definitions to be created, exchanged among systems, and maintained.

— A standard method by which Competency Definitions can be identified as globally unique among compliant systems and repositories.

— A standard method to mark a superseded or obsolete Competency Definition, and to point to a more current Competency Definition.

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

— Correspondence with the Learning Objects Metadata Standard developed by a parallel group.

2. References

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

IETF RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax," August 1998.

IEEE 1484-12:2002: Standard for Learning Object Metadata.

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

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

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

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

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

[Ed.: This may be informative; to be resolved] ISO/IEC 11404:1996, Information technology – Programming languages, their environments and system software interfaces – Language-independent datatypes.

3. Definitions

For purposes of this standard, the following terms and definitions apply. IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [A3], should be referenced for terms not defined in this Clause.

datatype: A property of distinct values, indicating common features of those values and operations on those values.

extended data element: An element of a data structure that is defined outside a standard and is permitted within an instance of the data structure.

LangString: A datatype that represents one or more character strings. A LangString value may include multiple semantically equivalent character strings, such as translations or alternative descriptions. See also: datatype.

competency: For this Standard, a competency is defined as any form of knowledge, skill, attitude, ability or educational objective that can be described in a context of learning, education or training.
Note—The word competency here is to be interpreted in the most broad sense to include educational objectives (those things that are sought) and competency or competencies (those things that are achieved). The word "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 word with nuance, for example limiting its use to skill and excluding knowledge or understanding.

smallest permitted maximum: For implementation-defined values, the smallest permitted maximum value. See also: clause 4.5.

value space: The set of values for a given datatype (ISO/IEC 11404:1996).
NOTE:--In this standard, a value space is typically enumerated outright, or defined by reference to another standard or specification.

3.1 Abbreviations and acronyms

[Editing note: Need to order and remove acronyms not in Standard]

ADL Advanced Distributed Learning

ISO International Standards Organization

JTC Joint Technical Committee

LTSC Learning Technology Standards Committee

RDCEO IMS Reusable Definition of Competency or Educational Objective

SCORM Shareable Courseware Object Reference Model

W3C World Wide Web Consortium

XML Extensible Mark-up Language

SPM: smallest permitted maximum

URI: Uniform Resource Identifier

URN: Uniform Resource Name

4. Conformance

Conformance to this standard is discussed in 4.1 – 4.5.

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

4.1 Data instances

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

4.5 Smallest permitted maximum values

In this Standard, smallest permitted maximum values are defined for:

— Items with multiple values: All applications that process RCD instances shall process at least that number of entries stated. In other words: an application may impose a maximum on the number of entries it processes for a data element with multiple values, but that maximum shall not be lower than the smallest permitted maximum value.

— Data elements with type CharacterString or LangString: All applications that process RCD instances shall process at least that length for the CharacterString value (either directly or contained in the LangString) of that data element. In other words: an application may impose a maximum on the number of characters it processes for the CharacterString value of that data element, but that maximum shall not be lower than the smallest permitted maximum value for the data type of the data element.

This standard defines smallest permitted maximum (SPM) values for data elements with data types that include bag, set, and characterstring. For these data elements, an implementation that conforms to this standard shall accept and process at least that number of entries or characters specified by the SPM for the element and may accept and process a larger number.

NOTES:

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

2—What "processing" means in the above depends on the nature of the application.

3—This standard does not define any provision for how and whether a system can process more than the SPM for a particular data element.

5. Conceptual model (informative)

5.1 Functional overview

This Standard defines an information model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning. In this specification, the word competency is used in a very general sense that includes skills, knowledge, tasks, and learning outcomes. This specification gives a way to formally represent the key characteristics of a competency, independent 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.

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

This Standard provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning pre-requisites, or as learning outcomes. The information model in this specification can be used to exchange these definitions between learning systems, human resource systems, learning content, competency or skills repositories, and other relevant systems. The RCD Standard provides unique references to descriptions of competencies or objectives for inclusion in other information models.

The RCD instances that conform to this Standard are intended for interchange by machines, but the information they contain is currently 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”) and does not address how competencies are to be assessed, 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 exchanged.

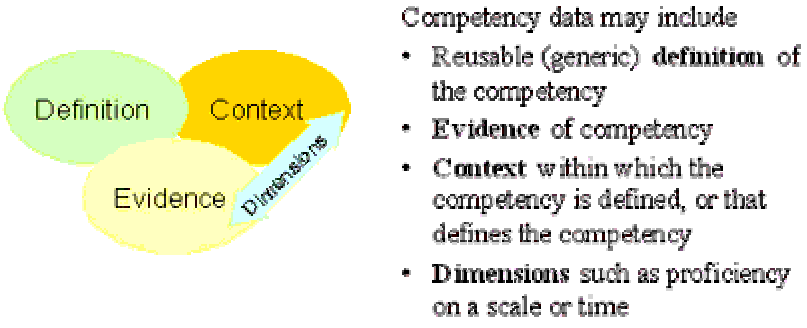

Figure 1—Reusable definition as part of competency data

5.2 Data model overview

The RCD data model is minimalist and extensible. It is purposely neutral with regard to models of competencies and the use of competencies. Competencies are defined and structured in many ways in different communities of practice (ACRL, CASAS, CPA, Mager, NOICC, O*Net, PASS, SCANS, TATS). This standard allows communities of practice to exchange information according to the model they use.

The data model contains the following mandatory elements:

a) Identifier: A globally unique label that identifies this Definition of Competency or Educational Objective. This identifier uses the same data elements as the Identifier element defined in the IEEE LOM standard, and consists of two sub-elements: Catalog and Entry. The Identifier is sufficient to reference the competency in any other system.

b) Title: A text label for the competency or objective. This is a short human-readable name for the competency. While the Identifier provides the definitive reference to the definition, it is typically unintelligible. The Title provides a convenient alternative readable form, but one which is not the definitive label. The Title may be repeated in multiple languages.

The other elements defined by the data model are optional:

c) Description: A human readable description of the competency. This is an optional unstructured (opaque) "text blob" meant to be interpretable only by humans. The Description may be repeated in multiple languages.

d) Definition: An optional structured description that provides a more complete definition of the competency or educational objective, usually using attributes taken from a specific model of how a competency or educational objective should be structured or defined. Typically, such models define a competency or educational objective in terms of a "statement, conditions, criteria," "proficiency, criteria, indicators," "standards, performance indicators, outcomes," "abilities, basic skills, content, process," and similar sets of statements.

e) Metadata: Optional embedded metadata that further describe the RCD.

Extensibility[DR1] can be achieved by defining a specific model structure within the RCD Definition element, or by including elements defined by the Learning Object Metadata (LOM) Standard in the Metadata portion. However, this standard does not define a specific extension mechanism for the data model. Implementers may create additional data models for competency data. Such models may be used to augment this model to support different communities of practice.

5.3 Taxonomies of Reusable Competency Definitions

There are various ways to classify competencies or educational objectives. This standard is intended to meet the simple need of referencing and cataloguing a competency or objective, not classifying it. Nonetheless, an implementation might want to include relation and classification information, which can be done through the embedding of optional metadata as specified in the data model.

Instances of Reusable Competency Definitions can also be referenced by the nodes in a tree or other graph representing a taxonomy or ontology of competencies.

6. Data model

This Clause defines the data elements of a RCD

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

NOTES:

1—The use of ISO/IEC 11404 notation in the synopses in 6.1 and 6.2 is for descriptive purposes only. A complete implementation of the operations defined in ISO/IEC 11404 is not required for conformance.

2—The ISO/IEC 11404 notation describes the semantics of the language-independent data types across all bindings (e.g., implementation of a data type as itself, its subtypes, its subclasses, and its specializations). For example, an ISO/IEC 11404 "record" may be implemented as an SQL table row, or as an XML complexType; an ISO/IEC 11404 "characterstring" may be implemented in an encoding (ISO 646, ASCII, ISO 8859-1, UTF-8, UTF-16, UTF-32, etc.) that supports the repertoire specified in the parameter to characterstring data type.

3—All examples in 6.1 and 6.2 are informative and do not endorse any particular binding.

4—The following language-independent data types used in this standard are defined in ISO/IEC 11404: bag, characterstring, record, set, state.

5—The labels for data elements and data types in the synopses in clauses 6.1 and 6.2 are for reference only. There is no requirement that an implementation use the exact same labels, as long as the data elements and data types are semantically equivalent.

6—This standard does not define a specific extension mechanism for the data model. Implementers may define binding that allow additional elements, or create additional data models for competency data. Such models may be used to augment this model to support different communities of practice.

6.1 Reusable Competency Definition

Synopsis

```
reusable_competency_definition :
record
(
  identifier :
    long_identifier_type,
  title :
    langstring_type(1000),
      // the parameter value is the SPM
  description :
    langstring_type(2000),
      // the parameter value is the SPM
  definition :
    record
    (
      model_source:
        characterstring(iso-10646-1),
          // SPM: 1000 characters
      statements:
        bag of statement_type,
          // SPM: 10 statement records in the bag
    ),
  metadata :
    record
    (
      rcd_schema:
        characterstring(iso-10646-1),
          // SPM: 4[illegible] characters[DR2]
      rcd_schema_version:
        characterstring(iso-10646-1),
          // SPM: 1000 characters
      additional_metadata:
        bag of any_type,
          // SPM: 10 of any type in the bag
    ),
)
```

Description

The components of `reusable_competency_definition` are defined in 6.1.1 – 6.1.5. Depending on the implementation, an instance of `reusable_competency_definition` shall include zero or more of the defined components.

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 bear no meaning. For ex-

ample, if the model includes three statements, then the order of these statements is not significant. They may appear in any order without loss of information.

Note—A binding may impose a particular ordering on RCD data instances that conform to that binding. Other than conformance to the binding, no significance is associated or should be inferred from that ordering requirement. In particular, the ordering of Statement elements in a Definition is undefined.

6.1.1 Identifier

Synopsis

```
identifier :
      long_identifier_type,
```

Description

A globally unique label that identifies this Reusable Competency Definition. The Identifier is sufficient to reference the competency definition in any other system.

Subclause 6.2.1 defines long_identifier_type.

NOTE— This identifier uses the same data elements as the Identifier element defined in the IEEE LOM standard, and consists of two sub-elements: Catalogue and Entry.

6.1.2 Title

Synopsis

```
title : bag of langstring_type(1000),
   // SPM: 20 instance of langstring_type in the bag
   // The parameter value is the SPM for the langstring
```

Description

A single mandatory text label for the competency or objective. This is a short human-readable name for the competency.

Subclause 6.2.1 defines langstring_type.

NOTES:

1—The Title may be repeated in multiple languages.

2— While the Identifier provides the definitive reference to the definition, it is typically unintelligible. The Title provides a convenient alternative readable form, but one which is not the definitive label. Examples: "English proficiency", "Schmiblick failure diagnostic level 4", "Demonstrates conflict resolution skills".

6.1.3 Description

Synopsis

```
description : bag of langstring_type(2000[DR3]),
   // SPM: 20 instance of langstring_type in the bag
   // The parameter value is the SPM for the langstring
```

Description

A human readable description of the competency. This is an optional unstructured (opaque) "text blob" meant to be interpretable only by humans.

Subclause 6.2.1 defines langstring_type.

NOTES:

1—The Description may be repeated in multiple languages.

2—The description is typically more explicative than the title. Examples: "Proficiency in written and spoken English and use of English for meaningful oral or written expression.", "Performance of level 4 diagnostic as specified in IETM #SCMBLK007"

6.1.4 Definition

Synopsis

```
definition :
   record
   (
      model_source :
         characterstring(iso-10646-1),
            // SPM: 1000 characters
      statements :
         bag of statement_type,
            // SPM: 10 statement records in the bag
   ),
```

Description

An optional structured description that provides a more complete definition of the competency or educational objective, usually using attributes taken from a specific model of how a compe-

tency or educational objective should be structured or defined. A definition shall contain at least one statement, and may multiple statements.

The components of `definition` are defined in 6.1.4.1 – 6.1.4.2.

NOTES:

1—Typically, the models that underlie the Definition define a competency or educational objective in terms of “statement, conditions, criteria”, “proficiency, criteria, indicators”, “standards, performance indicators, outcomes”, “abilities, basic skills, content, process”, and similar sets of statements.

2—The Definition element provides a structure for including an arbitrary collection of statements that determine a competency or educational objective. The author of a RCD is free to use the Definition element in the way that best describes the competency or objective.

3—The best practices guide and examples in informative Annex X will illustrate potential ways to express different types of competencies, objectives, and competency or objective models using the Definition element.

6.1.4.1 Model source

Synopsis

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

Description

Source of the model used for the competency definition. The characters in this string shall belong to the repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396.

NOTE—The value of Model Source should be specific enough to avoid conflict with other source names; therefore it is recommended that this value be a URI. If the value of Model Source is a URI, it may point to an actual document that defines the source formally. However, this is not required.Examples: "3-part-learning-objective", "http://foo.edu/ref/los.xml".

6.1.4.2 Statements

Synopsis

```
statements:
   bag of statement_type,
      // SPM: 10 statement records in the bag

statement_type = record  // SPM: 10 statement records in the bag
(
   statement_id:
      long_identifier_type,
```

```
      statement_name :
         characterstring(1000),
      statement_text : bag of langstring_type(1000),
         // SPM: 20 instance of langstring_type in the bag
         // the parameter value is the SPM for the langstring
      statement_token :
         vocabulary_type,
   ),
```

Description

Each record in `statements` is a description of a single characteristic of a definition. A record of type `statement_type` shall contain one or more elements.

The elements of a `statement_type` record are defined in 6.1.4.2.1 – 6.1.4.2.4

NOTE— Although no specific element of a statement is required, a statement must contain at least one or more of these elements. For example, a particular learning objective model requires a list of specific statement strings, each of which has a specific name, such as “Condition”, “Performance” and “Standard”; a Definition statement matching this model would use the elements Statement Name and Statement Text.

6.1.4.2.1 Statement ID

Synopsis

```
   statement_id :
       long_identifier_type,
```

Description

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

Subclause 6.2.6 defines `long identifier type`.

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

6.1.4.2.1 Statement name

Synopsis

```
   statement_name :
       characterstring,
```

Description

The `id` data element is a label for the interaction. This label shall be unique at least within the scope of the content object. Examples: "Condition", "Action", "Standard", "Outcome", "Criteria"

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

6.1.4.2.1 Statement text

Synopsis

```
statement_text : bag of langstring_type(1000),
   // SPM: 20 instance of langstring_type in the bag
   // the parameter value is the SPM for the langstring
```

Description

An optional unstructured textual description of those aspects of the competency referred to by the Statement Name. Example: "Given a set of integer numbers in the range 1 to 49".

NOTE— The Statement Text may be repeated in multiple languages.

6.1.4.2.1 Statement token

Synopsis

```
statement_token :
    vocabulary_type,

abulary_type = record

  source:
    characterstring(iso-10646-1),
      // SPM: 1000 characters
  value:
      characterstring(iso-10646-1),
      // SPM: 1000 characters
[DR4]),
```

Description

An optional vocabulary token, along with an identifier of the source of the vocabulary. This allows the use of controlled terms (vocabularies) instead of, or along with, free-form statement text.

The source element is an indication of the source of the value. It may be a URI that identifies a formal vocabulary definition. For example, "http://www.vocabularies.org/OSList".

The value element is the actual token value, from a list of tokens defined in the source. For example, "MRS_15"

NOTE 1— This approach to controlled terms (vocabularies) follows that used in metadata standards such as IEEE LOM. The token is just a string; it does not have to be a human language word and does not have to be meaningful. 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 URL to a human or machine-readable description of the vocabulary terms.

NOTE 2—This standard does not specify how vocabularies are created, assigned, or resolved.

6.1.5 Metadata

Synopsis

```
metadata :
  record
  (
    rcd_schema:
      characterstring(iso-10646-1),
        // SPM: 1000 characters
    rcd_schema_version:
      characterstring(iso-10646-1),
        // SPM: 1000 characters
    additional_metadata:
      bag of any type,
        // SPM: 10 of any type in the bag
  ),
```

Description

Embedded metadata about this RCD.

The components of `metadata` are defined in 6.1.5.1 – 6.1.5.3

6.1.5.1 RCD schema

Synopsis

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

Description

Describes the schema that defines and controls this RCD data.instance.

NOTES

1—If this element is omitted then a value of "IEEE 1484.20" should be assumed (i.e., default interpretation of elements). Different values may be used to signal application profiles but must not be used to replicate the purpose of other elements such as Model.

2—This element does not describe the schema of the embedded metadata defined in 6.1.5.3. Every instance of embedded metadata, if any, should include its own schema description.

6.1.5.2 RCD schema version

Synopsis

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

Description

Describes the version of the schema described by `rcd_schema`.

NOTE—If this element is omitted then a value of "1.0" should be assumed.

6.1.5.3 Additional metadata

Synopsis

```
additional_metadata :
   bag of any_type, // SPM: 10 of any type in the bag
```

Description

Optional additional embedded Metadata describing this RCD.

If a metadata record is included, it is recommended that this record conform to IEEE 1484-12.1-2002: Standard for Learning Object Metadata (IEEE LOM). In such conforming records, the version of the Metadata Specification standard is given in the meta-metadata element of the metadata record.

NOTES:

1—Useful metadata defined in the IEEE LOM include additional identification as an entry in one or more catalogues, information about the author, the creation date, and the coverage (in the sense of the Dublin Core as adopted by the IEEE LOM.) The Relation element may be used to relate a definition to a prior version of the definition, and one or more Classification elements may be used to indicate where this particular definition fits in a taxonomy of competencies or educational objectives.

2—More than one metadata record is allowed in the bag, but if there is more than one record each record should conform to a different metadata specification. An implementation must accept any metadata record that it cannot interpret, but it is not required to interpret such metadata records.

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

6.2 Auxiliary data types

The following data types are used in conjunction with the data elements described in Clause 6.1.

6.2.5 LangString type

Synopsis

```
type langstring_type(length) =
   record
   (
      language :
         language_type,
      string :
         characterstring(iso-10646-1)
            // SPM: The length parameter
   );
```

Description

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

Examples

The following are three examples of localized strings: `"Information Technology"` in French, `"localization"` in British English, and `"xxx"` in Japanese hiragana.

```
( "fr", "Technologies de l'information" )
( "en-GB", "localisation" ),
( "jp-JP-jisx208", "xxx" ),
```

[5]

6.2.5.1 Language

Synopsis

```
language :
   characterstring(iso-646),
      // SPM: 250 characters
```

Description

The `language` data element specifies the language of the localized string. The format of this data type is a character string consisting of a required language code followed by multiple, optional, hyphen-prefixed subcodes (see examples below).

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

- 2-letter codes are defined by ISO 639–1.
- 3-letter codes are defined by ISO 639–2.

641 – The value prefix `"i"` is reserved for registrations defined by the Internet As-
642 signed Numbers Authority (IANA).
643 – The value prefix `"x"` is reserved for private use.

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

645 – 2-letter subcodes are ISO 3166–1 alpha-2 country codes.
646 – Subcodes of from 3 to 8 letters are registered with IANA.

647 Rules for additional subcodes are unspecified.

648 NOTE—The language code is normally given in lower case and the subcodes (if any) in upper
649 case. However, the values are case insensitive.

650 **Examples**

651 `"en-GB"`
652 `"de"`
653 `"fr-CA"`
654 `"it"`
655 `"grc"` (Ancient Greek, until 1453)
656 `"en-US-philadelphia"`
657 `"eng-GB-cockney"`
658 `"map-PG-buin"` (Austronesian - Papua New Guinea Buin)
659 `"gem-US-pennsylvania"`

660 ### 6.2.5.2 String

661 **Synopsis**

```
662 string :
663    characterstring(iso-10646-1),
664       // SPM: The length parameter
```

665 **Description**

666 The `string` data element contains the text of the localized string.

667 6.2.6 Long identifier type

668 **Synopsis**

```
669 type long_identifier_type =
670    record (
671       catalog: characterstring(iso-10646-1),
672          // SPM: 4000 characters
673       entry: characterstring(iso-10646-1)
674          // SPM: 4000 characters
675    );
```

Description

This data type is an identifier (a label) that is intended to be unique within the context of usage of the RCD. The catalog and entry values shall conform to the syntax for Uniform Resource Identifiers (URIs) as defined by RFC 2396. 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 conform to the syntax for Uniform Resource Identifiers (URIs) as defined by RFC 2396.

NOTES

1—This standard recommends that if an application profile or binding specifies a concatenated format for the identifier, the result be in the form of a globally unique identifier in the form of a Uniform Resource Name (URN) (see RFC 2141 [A5]).

[Question: Should there be a note recommending that in practice catalog and entry lengths should be limited, so that the total length of a concatenated identifier would be no more than 4000 characters?]

Annex A

(informative)

Bibliography

[Editing note: To be reformatted to use [A1] ref syntax like IEEE 1484.11.1, and reordered/renumbered in same order as in text; anything not referenced in text to be removed]

[A2] IEEE 1484.11.2–2003, Standard for Learning Technology—ECMAScript Application Programming Interface for Content to Runtime Services Communication.

[A3] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

[A5] RFC 2141, "URN Syntax," Network Working Group, May 1997.

IMSRDCEO IMS Global Learning Consortium, Inc. (2002). IMS Reusable Definition of Competency or Educational Objective, Version 1.0

A2A http://www.aligntoachieve.org (Align to Achieve)

ACRL http://www.ala.org/acrl/ilstandardlo.html (Association of College and Research Libraries. Information Literacy Competency Standards for Higher Education: Standards, Performance Indicators, and Outcomes)

CASAS http://www.casas.org/

CPA http://www.cpavision.org/poll/corecomp.cfm (Core Competencies for CPAs)

HR-XML http://www.hr-xml.org/ (HR-XML Consortium)

IMSBUND http://www.imsglobal.org/implementationhandbook/imspack_handv1p0.html (Using IMS Content Packaging to Package Instances of LIP and Other IMS Specifications)

IMSMD http://www.imsglobal.org/metadata/ (IMS Metadata Specification)

IMSPLID http://www.imsglobal.org/implementationhandbook/imsrid_handv1p0.html, IMS Persistent, Location-Independent, Resource Identifier Implementation Handbook version 1.0

LSDA http://www.lsda.org.uk/ (Learning and Skills Development Agency)

Mager Robert Mager, 1984. Preparing Instructional Objectives, 2nd Edition. Lake Pub. Co., Belmont, CA.

NOCN http://www.nocn.org.uk/ (National Open College Network)

NOICC http://www.academicinnovations.com/noicc.html (National Occupational Information Coordinating Committee: High School Student Competencies and Indicators)

O*NET http://online.onetcenter.org/ (or http://www.access.gpo.gov/o_net/datadict/datadict.pdf)

Ostyn http://ltsc.ieee.org/doc/wg20/CompDefInit.doc (Base document from P1484.20)

PASS http://www.ous.edu/pass/standards/admission.html (Oregon Proficiency-based Admissions Standards System)

SCANS http://www.tier.net/tcenters/scans.htm (Secretary's Commission on Achieving Necessary Skills: Competencies)

SCORM http://www.adlnet.org (ADL SCORM)

TATS http://www.adtdl.army.mil/atdls.htm (Total Army Training System)

739

Annex B

(informative)

Sample 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.

This example illustrates existing practice using an XML schema defined by the IMS Global Learning Consortium. This XML Schema instance uses a different label for the root data element ("rdceo" instead of "rcd"). It is not required that an implementation of this standard use the exact same labels for data element or types labels as those used in this standard, as long as the elements and types themselves are semantically equivalent.

```
<?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>
```

```
765         <xs:complexType>
766           <xs:sequence>
767             <xs:element ref="identifier" minOccurs="1" maxOccurs="1"/>
768             <xs:element ref="title"/>
769             <xs:element ref="description" minOccurs="0"/>
770             <xs:element ref="definition" minOccurs="0" maxOccurs="unbounded"/>
771             <xs:element ref="metadata" minOccurs="0"/>
772             <xs:sequence minOccurs="0">
773               <xs:group ref="extelement"/>
774             </xs:sequence>
775           </xs:sequence>
776           <xs:anyAttribute namespace="##other" processContents="strict"/>
777         </xs:complexType>
778       </xs:element>
779       <xs:element name="langstring">
780         <xs:annotation>
781           <xs:documentation>A string in a human language</xs:documentation>
782         </xs:annotation>
783         <xs:complexType>
784           <xs:simpleContent>
785             <xs:extension base="xs:string">
786               <xs:anyAttribute namespace="##other" processContents="strict"/>
787             </xs:extension>
788           </xs:simpleContent>
789         </xs:complexType>
790       </xs:element>
791       <xs:element name="title">
792         <xs:annotation>
793           <xs:documentation>A title for the definition</xs:documentation>
794         </xs:annotation>
795         <xs:complexType>
796           <xs:sequence>
797             <xs:element ref="langstring" maxOccurs="unbounded"/>
798             <xs:sequence minOccurs="0">
799               <xs:group ref="extelement"/>
800             </xs:sequence>
801           </xs:sequence>
802           <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>
          <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>
```

```
879           <xs:documentation>Used for statements with free-form text</xs:documentation>
880         </xs:annotation>
881         <xs:complexType>
882           <xs:sequence>
883             <xs:element ref="langstring" maxOccurs="unbounded"/>
884             <xs:sequence minOccurs="0">
885               <xs:group ref="extelement"/>
886             </xs:sequence>
887           </xs:sequence>
888         </xs:complexType>
889       </xs:element>
890       <xs:element name="source">
891         <xs:annotation>
892           <xs:documentation>Source identification for a vocabulary token</xs:documentation>
893         </xs:annotation>
894         <xs:complexType>
895           <xs:simpleContent>
896             <xs:extension base="xs:string">
897               <xs:anyAttribute namespace="##other" processContents="strict"/>
898             </xs:extension>
899           </xs:simpleContent>
900         </xs:complexType>
901       </xs:element>
902       <xs:element name="value">
903         <xs:annotation>
904           <xs:documentation>A vocabulary token</xs:documentation>
905         </xs:annotation>
906         <xs:complexType>
907           <xs:simpleContent>
908             <xs:extension base="xs:string">
909               <xs:anyAttribute namespace="##other" processContents="strict"/>
910             </xs:extension>
911           </xs:simpleContent>
912         </xs:complexType>
913       </xs:element>
914       <xs:element name="metadata">
915         <xs:annotation>
916           <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 ab-
    sent.</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>
```