

1 **Proposed**  
2 **Draft Standard for Learning Technology—**  
3 **Simple Reusable Competency Map**

4 *DOCUMENT STATUS: INCOMPLETE DRAFT IN PROGRESS, SUBJECT TO*  
5 *RADICAL CHANGE.*  
6 *THIS ROUGH DRAFT IS CIRCULATED FOR FEEDBACK, COMMENTS,*  
7 *CRITIQUE AND IMPROVEMENTS.*  
8 *CO-AUTHORS ARE WELCOME – CONTACT CLAUDE AT OSTYN.COM*  
9

10

11 This document is a draft for a Standards project to be submitted to the Learning  
12 Technology Standards Committee of the IEEE Computer Society

13

14 Copyright © 2005 by Claude Ostyn

15 All rights reserved. This document is an unapproved draft of a document intended for submis-  
16 sion as a possible IEEE Standard and is subject to change without notice. Permission is hereby  
17 granted for Ostyn Consulting customers and partners, and for participants in industry Standard  
18 initiatives, to reproduce and redistribute this document for purposes of Standardization ser-  
19 vices. If this document is to be submitted to any Standards body or included in such a submis-  
20 sion, notification shall be given to Claude Ostyn. Other entities seeking permission to repro-  
21 duce all or any portion of this document must contact Ostyn Consulting at [www.ostyn.com](http://www.ostyn.com) for  
22 the appropriate license.

23 **Abstract**

24 This Standard defines an information model for simple, reusable competency maps to be used  
25 for describing, referencing, and exchanging data about the relationships between competen-  
26 cies, primarily in the context of online and distributed learning. The information model allows  
27 the representation of relationships between competencies or complementary aspects of compe-  
28 tency, such as they have often been captured in competency models. In this Standard, the word  
29 competency is used in a very general sense that includes skills, knowledge, attitude, and learn-  
30 ing outcomes. This Standard references the IEEE P1484.20.1 Reusable Competency Defini-  
31 tion (RCD) Standard (in draft status at of this writing). The RCD Standard specifies how any  
32 arbitrary competency definition can be captured using Standard data model. In this Standard,  
33 the relationships between RCDs are represented is a map in which each node may reference a  
34 RCD. Simple "roll up" rules are specified to allow representation of how sub-competencies  
35 specified in the map can "roll up" to a broader competency.

36 **Keywords:**

37 competency, competency definition, objective, competency map, competency taxonomy, hi-  
38 erarchy, directed acyclic graph

39 *[NOTE: Information about IEEE LTSC P1484.20 can be found at:*

40 <http://www.ieeeltsc.org/wg20Comp/>

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

42 **Introduction**

43 (This introduction is not a part of xxxx, Draft Standard for Learning Technology—Data  
44 Model for Reusable Competency Map.)

45 This Standard defines a data model for describing, referencing, and exchanging simple lists or  
46 hierarchical maps representing the relationships between reusable competency definitions,  
47 primarily in the context of online and distributed learning.

48 **Participants**

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

<i>TBD Chair</i>		
(other names to be updated)		

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

52 **Acknowledgements**

53 TBD.

54	<b>Contents</b>	
55	<b>1. Overview</b> .....	<b>1</b>
56	1.1 Scope .....	2
57	Purpose .....	3
58	<b>2. References</b> .....	<b>4</b>
59	<b>3. Definitions</b> .....	<b>4</b>
60	3.1 Abbreviations and acronyms .....	5
61	<b>4. Conformance</b> .....	<b>6</b>
62	4.1 Data instances .....	6
63	4.5 Smallest permitted maximum values.....	6
64	<b>5. Conceptual model (informative)</b> .....	<b>7</b>
65	5.1 Functional overview .....	7
66	5.1.1 Modeling.....	8
67	5.1.2 Directed acyclic graph (DAG) map shape .....	8
68	5.1.3 Map extension by reference .....	10
69	5.1.5 Example applications .....	13
70	5.2 Data model overview.....	15
71	<b>6. Data model</b> .....	<b>16</b>
72	6.1 Reusable Competency Map .....	17
73	6.1.1 Identifier.....	18
74	6.1.2 Title .....	18
75	6.1.3 Description.....	18
76	6.1.4 SRCM schema label.....	19
77	6.1.5 SRCM schema version.....	19
78	6.1.6 Referential.....	20
79	6.1.7 Metadata.....	20
80	6.1.8 Graph.....	21
81	6.1.9 Extensions.....	22
82	6.2 Data types.....	23
83	6.2.1 Node type.....	23
84	6.2.1 Any type.....	30
85	6.2.2 LangString type .....	30
86	6.2.3 Long identifier type.....	32
87	6.2.3 Short identifier type.....	32
88	<b>Annex B (informative) Sample XML Binding</b> .....	<b>34</b>

# 89 **Draft Standard for Learning Technology—** 90 **Simple Reusable Competency Map**

## 91 **1. Overview**

92 This Standard defines an information model for simple, reusable competency maps to be used  
93 for describing, referencing, and exchanging data about the relationships between competen-  
94 cies, primarily in the context of online and distributed learning. The information model allows  
95 the representation of relationships between competencies or complementary aspects of compe-  
96 tency, such as they have often been captured in competency models. In this Standard, the word  
97 competency is used in a very general sense that includes skills, knowledge, attitude, and learn-  
98 ing outcomes. This Standard references the IEEE P1484.20.1 Reusable Competency Defini-  
99 tion (RCD) Standard (IEEE project in draft status at of this writing). The RCD Standard speci-  
100 fies how any arbitrary competency definition can be captured using Standard data model. In  
101 this Standard, the relationships between RCDs are represented as a map in which each node  
102 may reference a RCD. Simple "roll up" rules are specified to allow representation of how sub-  
103 competencies specified in the map can "roll up" to a broader competency.

104 The Standard specifies that the must be a directed acyclic graph. A related Standards project  
105 may specify other map shapes, such as a general directed graph, in order to allow the represen-  
106 tation of competency models of arbitrary complexity, but such graph shapes and representa-  
107 tions are outside the scope of this Standard. This Standard enables interoperability among  
108 learning systems and competency management systems that deal with competency informa-  
109 tion by providing a means for them to refer to common competency definitions and to model  
110 the relationships between those common definitions through the use of interoperable maps.  
111 This Standard is not intended to define ontologies or to duplicate semantic mapping Standards  
112 such as topic maps; however, it allows the representation of competency maps as directed  
113 acyclic graphs generated by processes that traverse ontologies or topic maps.

114 Different communities of practice tend to use different models to represent the competencies  
115 required for tasks, readiness for a job, or mastery of a subject domain, but increasingly they  
116 often need to exchange and manage competency information with each other. The information  
117 model in this Standard can be used to exchange these models between learning systems, hu-  
118 man resource systems, learning content, competency or skills repositories, and other relevant  
119 systems. The SRCM that conform to this Standard are intended for interchange by machines,  
120 just like the RCD that conform to the RCD Standard.

## 121 **1.1 Scope**

122 This Standard shall specify the mandatory and optional data elements that constitute a Simple  
123 Reusable Competency Map as used in a Learning Management System, or referenced in a  
124 Competency Profile. This Standard is intended to satisfy the following objectives:

125 — Provide a Standardized data model to represent the parent, child or sibling relation-  
126 ships that may exist between Reusable Competency Definitions.

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

128 — Provide a Standardized way to represent the result of grouping and decomposing  
129 competency information captured in Reusable Competency Definitions. Nodes that represent  
130 groupings or products of decomposition can be mapped to new or existing Reusable Compe-  
131 tency Definitions.

132 — Provide a unique identifier as the means to unambiguously reference a Simple Reus-  
133 able Competency Map, regardless of the setting in which this Competency Map is stored,  
134 found, retrieved, or used. For example, metadata that describe learning content may contain a  
135 reference to one or more Simple Reusable Competency Maps that provide the context for a  
136 classification of the learning objectives for the content.

137 — Provide a Standardized data model for additional information about a Simple Reusable  
138 Competency Map, such as a title, description, and source, compatible with other emerging  
139 learning asset metadata Standards

140 — Given any Reusable Competency Definition and a Simple Reusable Competency Map  
141 that contains a reference to this Reusable Competency Definition, provide the data structure  
142 that allows the discovery of related competency definitions through the structure of the map.

143 — Provide a Standardized data model for additional information about relationships be-  
144 tween groups of competencies represented by a Reusable Competency Map Definition, such  
145 as relative weights, rollup rules governing how component competencies can be considered to  
146 add up to higher level competencies, and proficiency levels required to assert mastery of a  
147 competency in the context of a hierarchy of competencies defined by the map.

148 — Provide a controlled vocabulary to express how Simple Reusable Competency Maps  
149 are semantically related in a list or hierarchical model.

150 This Standard specifically does not cover:

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

153 — Quality and accuracy in the data itself, although it will describe recommended best  
154 practices. For example, this Standard does not cover the quality or validation of the various  
155 component competencies that make up a higher level competency, or the relevance of the Re-  
156 usable Competency Definitions referenced by a node in a Reusable Competency Map.

- 157 — The processes by which a SRCM is created, generated, maintained or published.
- 158 — How the relationships between competencies are stored in a database or learning man-  
159 agement system.
- 160 — Any representation of relationships between competencies that requires a general  
161 graph rather than a list or directed acyclic graph.
- 162 — Mapping or references to data objects other than Reusable Competency Definitions or  
163 other Simple Reusable Competency Maps.
- 164 — Certification data models and how they may be referencing SRCMs
- 165 *[Note (to be moved somewhere else). However, Certification records may reference Simple*  
166 *Reusable Competency Maps along with Competency Definitions. For example, an accredited*  
167 *authority may grant certificates that acknowledge that an individual meets the requirements*  
168 *for a particular competency after walking an associated Simple Reusable Competency Map to*  
169 *identify requirements for component competencies that add up to the target competency.]*
- 170 — Individual competency records, as would be found in the competency profiles of indi-  
171 viduals or groups, and how they may be referencing SRCMs.
- 172 *[Note (to be moved somewhere else) However, such records can include references to specific*  
173 *Simple Reusable Competency Maps along with references to Reusable Competency Defini-*  
174 *tions. For example, a competency profile for an individual may include a collection of certifi-*  
175 *cates which in turn reference Competency Definitions, and that collection may be discovered*  
176 *by inspecting a Simple Reusable Competency Map that references any of those Competency*  
177 *Definitions. Skill gaps may be discovered by finding the map nodes for which no evidence of*  
178 *competency exists.]*
- 179 — Confidence or trustworthiness of competency records; confidence in results of sum-  
180 mation that includes rolling up proficiency for multiple competencies shown by a map to be  
181 related, where some competency records may vary in trustworthiness..

## 182 **Purpose**

183 The purpose of this Standard is to define a universally acceptable Simple Reusable Compe-  
184 tency Map Definition model to allow the creation, exchange and reuse of hierarchical maps  
185 representing the relationships between Competency Definition in applications such as Learn-  
186 ing Management Systems, Competency or Skill Gap Analysis, Learner and other Competency  
187 profiles, and so on. The Standard is needed because there are currently many definitions of the  
188 terms "Competency Model", "Competency taxonomy", "Learning Objective", "Competency"  
189 and "Skill", and very little agreement between how those definitions can be used to define re-  
190 usable data models to support automation and computer-assisted discovery. This Standard  
191 uses a general definition that can be applied to any scale of competency hierarchy, from shal-  
192 low ad-hoc taxonomies used in an assessment to deep formal hierarchies representing a do-

193 main, while conserving the same data model regardless of how strictly a particular organiza-  
194 tion or institution requires the data to be formulated. This Standard also addresses the follow-  
195 ing needs:

196 — A common data model that allows the building of various ad hoc or formal models,  
197 inventories, hierarchies and mappings of Competency Definitions.

198 — A Standard that allows persistent, long lived Simple Reusable Competency Maps to  
199 be created, exchanged among systems, and maintained.

200 — A Standard data element by which  
201 Simple Reusable Competency Maps can be identified as globally unique among compliant  
202 systems and repositories.

203 — A common data model to represent the result of aggregating or decomposing compe-  
204 tency information captured in RCDs through the use of additional RCDs representing the  
205 products of the decomposition, and to capture assertions about the hierarchical relationship  
206 between the original RCD and the subsidiary RCDs in the form of a Simple Reusable Compe-  
207 tency Map.

208 — A common data model for the descriptive or cataloging metadata that give a reusable  
209 Simple Reusable Competency Map its value in a reuse environment. Such metadata may typi-  
210 cally include the publisher of the Simple Reusable Competency Map, validation information,  
211 and other descriptive information useful to locate an existing competency map in a repository  
212 catalog or collection index.

213 — Correspondence with the IEEE Learning Objects Metadata Standard (IEEE  
214 1484.12.1) developed by a parallel group.

## 215 **2. References**

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

219 (P)1484.20.1 Standard for Learning Technology – Reusable Competency Definitions

220 (TBD: 1484.12.1 LOM)

## 221 **3. Definitions**

222 For purposes of this Standard, the following terms and definitions apply. IEEE 100, *The Au-*  
223 *thoritative Dictionary of IEEE Standards Terms*, Seventh Edition [A3], should be referenced  
224 for terms not defined in this Clause.

225 (TBD)

226 **LangString:** A datatype that represents one or more character strings. A LangString  
227 value may include multiple semantically equivalent character strings that represent  
228 translations of the same meaning into different languages. See also: datatype.

229 **competency:** For this Standard, a competency is defined as any form of knowledge, skill,  
230 attitude, ability or learning outcome that can be described in a context of learning, edu-  
231 cation or training.

232 **Note—**The word competency here is to be interpreted in the most broad sense to include  
233 educational objectives (those things that are sought) and competency or competencies  
234 (those things that are achieved). The word “competency” is also used to include all  
235 classes of things that someone, or potentially something, can be competent in, although  
236 some communities of practice use the word with nuance, for example limiting its use to  
237 skill and excluding knowledge or understanding. In learning contexts, a competency to  
238 be achieved is often named “objective”.

239 **smallest permitted maximum:** For implementation-defined values, the smallest permit-  
240 ted maximum value. See also: clause 4.5.

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

### 244 3.1 Abbreviations and acronyms

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

246 ADL Advanced Distributed Learning

247 IMS IMS Global Learning Consortium

248 ISO International Standards Organization

249 JTC Joint Technical Committee

250 LTSC Learning Technology Standards Committee

251 RCD Reusable Competency Definition

252 RDCEO IMS Reusable Definition of Competency or Educational Objective

253 SCORM Shareable Courseware Object Reference Model

254 W3C World Wide Web Consortium

255 XML Extensible Mark-up Language

256 SPM smallest permitted maximum

257 URI: Uniform Resource Identifier

258 URN: Uniform Resource Name

## 259 **4. Conformance**

260 Conformance to this Standard is discussed in 4.1 – 4.5.

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

263 Note. Since this Standard defines a data model but not a specific binding, and system confor-  
264 mance cannot be defined without one or more binding, conformance of systems is outside the  
265 scope of this Standard.

### 266 **4.1 Data instances**

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

### 268 **4.5 Smallest permitted maximum values**

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

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

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

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

285 NOTES:

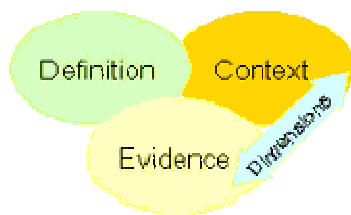
286 1—The intent is for the SPM values to cover most cases.

287 2—What "processing" means in the above depends on the nature of the application.

288 3—This Standard does not define any provision for how and whether a system can process more  
289 than the SPM for a particular data element.

## 290 5. Conceptual model (informative)

### 291 5.1 Functional overview



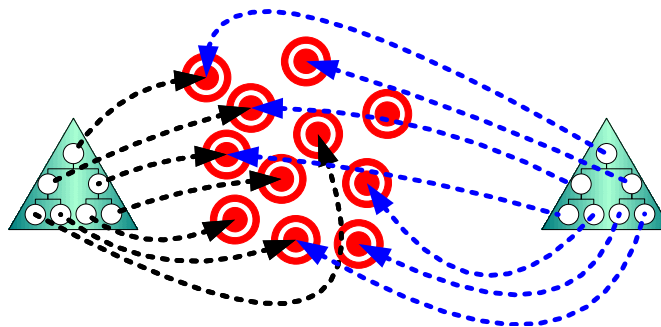
Competency data may include

- Reusable (generic) definition of the competency
- Evidence of competency
- Context within which the competency is defined, or that defines the competency
- Dimensions such as proficiency on a scale or time

292

293 **Figure 1—Reusable competency definitions capture only a part of the data that define a**  
294 **competency**

295 Reusable Competency Definitions (RCD), as defined in IEEE P1484.20.x, may used to cap-  
296 ture a competency definition at any level of specificity, from the most precise to the most gen-  
297 eral. The more specific a RCD is, the less reusable it is. Often a less precise definition is very  
298 useful, especially when trying to compare competency data between different communities of  
299 practice. For example, as a tourist you are typically considered competent to drive a car in a  
300 foreign country, even though the details of the competency model for driving competency  
301 may be very different between countries. In that case, the gross competency definition is good  
302 enough and going into details would impair commerce when it comes to renting a car. In  
303 other cases, though, you do want to be able to reference a competency in the context of a spe-  
304 cific model that corresponds to the expectations or requirements of a specific community of  
305 practice. Such a model can be represented by a competency map.



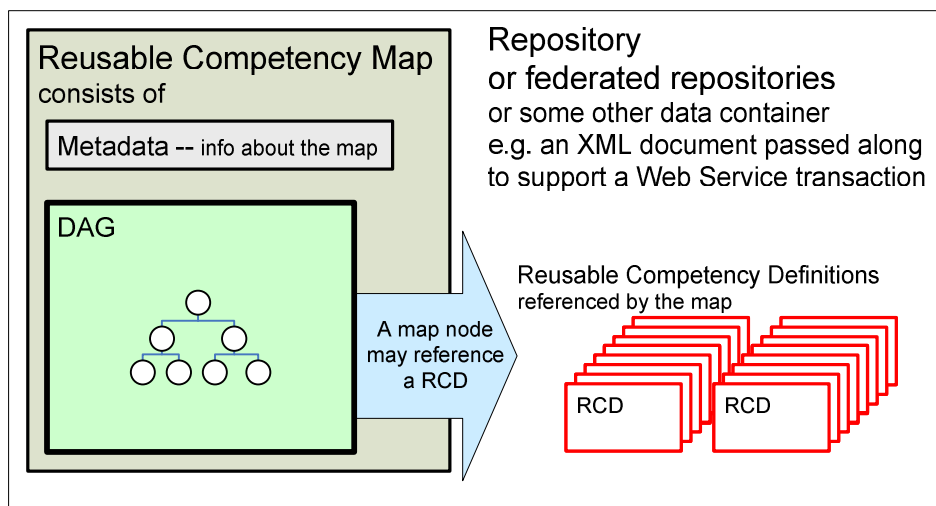
306

307 **Figure 2— Competency maps may represent different models of the same competency**

### 308 5.1.1 Modeling

309 The SRCM data model is minimalist and extensible. It is purposely neutral with regard to  
 310 models of competencies and the use of competencies. Competencies and competency models  
 311 are defined and structured in many ways in different communities of practice. This Standard  
 312 allows many communities of practice to exchange useful information regardless of the model  
 313 they use, as long as the model can be represented or exported in a shape that can be repre-  
 314 sented according the Standard.

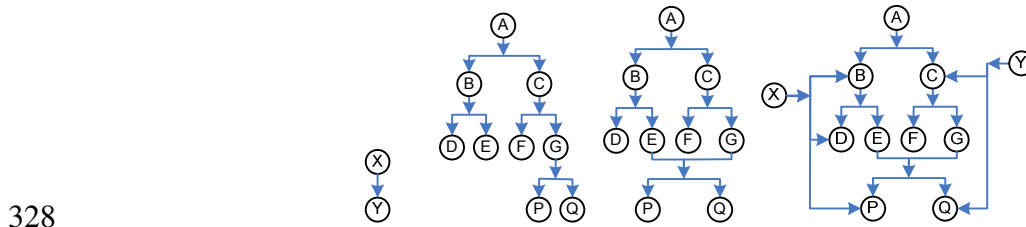
### 315 5.1.2 Directed acyclic graph (DAG) map shape



316

317 **Figure 3— Competency maps as a graph with references to RCDs**

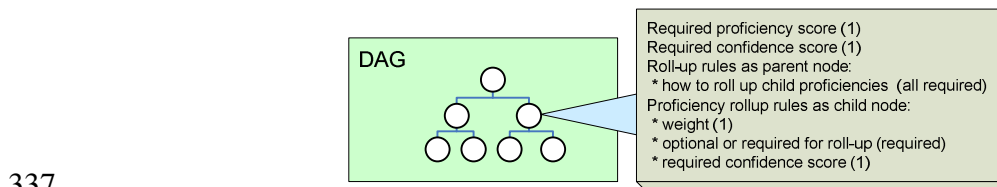
318 The Standard specifies the shape of a SRCM as a directed acyclic graph (DAG). A DAG is a  
 319 hierarchical collection of nodes that implies containment. If node A has children B and C, it  
 320 implies that A contains B or C. In competency terms, this would imply that competency A  
 321 may be decomposed into sub-competencies or competency facets B and C, or that B and C  
 322 contribute to A. In reality, many so-called existing competency “taxonomies” are not true tax-  
 323 onomies, because the same competency components tend to appear more than once in the  
 324 model hierarchy. The Standard allows this to be represented in a DAG. The DAG allows a  
 325 node to have more than one parent, as long as the parent cannot be a descendant of the node.  
 326 Figure 4 shows four examples of DAGs, from simple to more complex. In the figure, arrows  
 327 represent a parent-child relationship, with the arrows going from parent to child.



329 **Figure 4 — Some DAG topologies with different levels of complexity.**

330 A simple topology for a DAG is a tree with a single root node and no commonality between  
 331 sub-trees within the tree. A more complex topology may specify common children for more  
 332 than one node, or more than one origin node. For example, in the rightmost graph in Figure 4,  
 333 A, X and Y represent different competencies that have certain component competencies in  
 334 common.

335 Nodes in a map can have specific rules useful for modeling. Different rules can apply to indi-  
 336 vidual nodes.



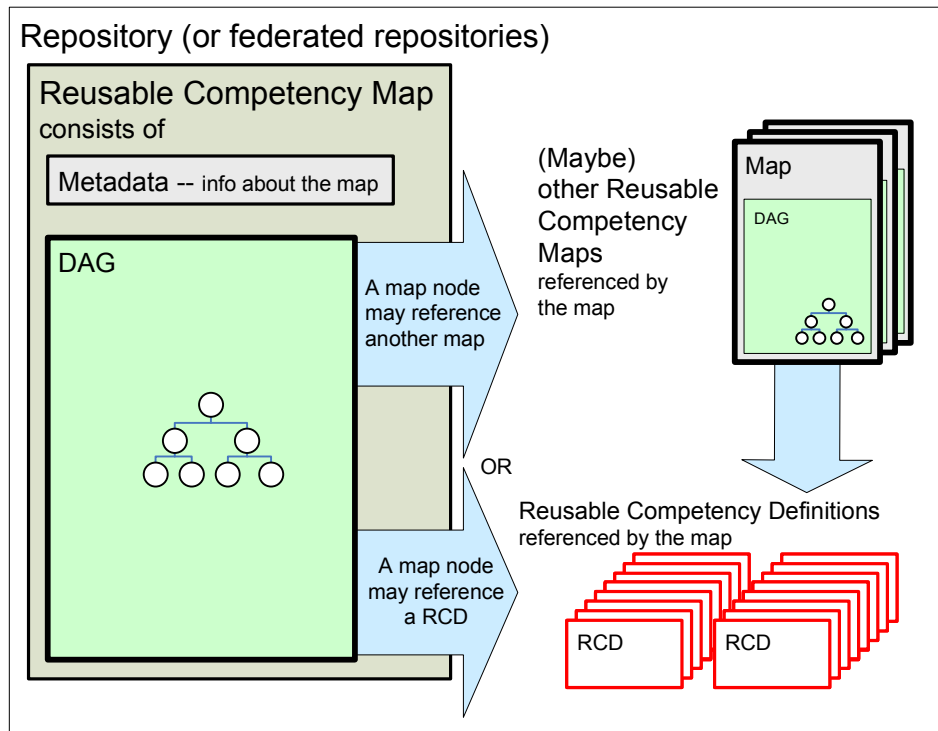
338 **Figure 4— Rules may be associated with nodes to define interpretations of the map**

339 The fact that a rule is defined in the Standard does not imply that it is required in all applica-  
 340 tions. For example, the required score threshold rules in Figure 4 may be required in an appli-  
 341 cation profile for an assessment request, but may not be important when reporting a compe-  
 342 tency inventory for a person. In any case, the Standard defines a default value for each rule.  
 343 For example, by default it is assumed that proficiency is required for all "children" of a de-  
 344 fined competency in order to achieve proficiency for the "parent".

345 An application of the competency map may add other rules, but this is outside the scope of  
 346 this Standard. For example, a competency map may be used to specify how to roll up profi-  
 347 ciency data throughout a collection of competency records that reference the RCDs that are  
 348 represented in the map. Each competency record typically references a RCD and includes a  
 349 data element for proficiency status and maybe proficiency score. An application that builds or  
 350 maintains a monolithic personal competency profile, rather than using cross references, might  
 351 merge data from those competency records with the structure of the competency map, in  
 352 which case the resulting data structure would include not only the nodes of the competency  
 353 map, but also data elements such as proficiency status for each node.

354 **5.1.3 Map extension by reference**

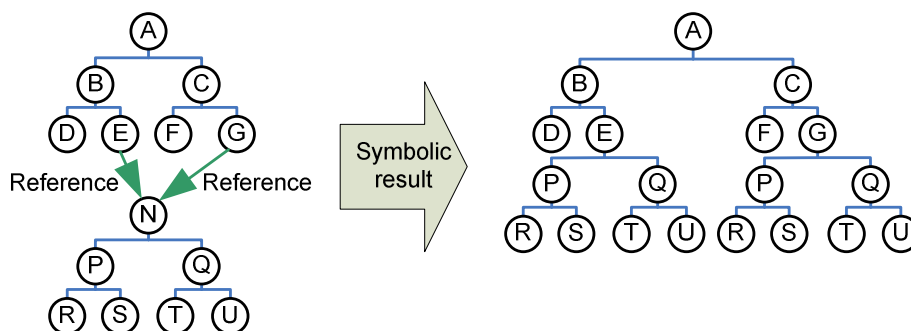
355 Maps may be symbolically merged by references to other maps. Any node in a map may ref-  
 356 erence a RCD, another map, or both. This is useful to allow the association of a more detailed  
 357 map with a RCD. For example, a list of competency definitions may be extended by referenc-  
 358 ing maps that provide detailed breakdowns the components of those competency definitions as  
 359 assumed by the creator of the list.



360

361 **Figure 6 — Competency maps may also include other maps by reference**

362 Application profiles may put constraints on the shapes that can be merged because of the  
 363 complexity of the resulting map. Application profiles may also specify that when maps are  
 364 merged, the referencing node is treated as a functional boundary, and that references to other  
 365 maps are followed only on demand in a more costly operation.



366

367 **Figure 7— Extension of a DAG map by reference to another DAG map**

## 368 **Map extension rules and constraints**

369 If a node in a DAG shaped map references another DAG shape map with a single origin  
 370 ("root") node, the root node of the DAG is subsumed in the referencing node. A node in a  
 371 DAG competency map can therefore represent any other DAG competency map. Several con-  
 372 strains do apply in order to retain sanity:

373 -- The result cannot be a cyclical topology. In other words, a node cannot become a descen-  
 374 dant of itself as a result of merging graphs, because this violates the basic acyclic constraint of  
 375 a DAG. Implementations must treat any node that would violate this constraint as a leaf node  
 376 and ignore its reference to another map. For example, an implementation may use the same  
 377 method used by a web site spider to avoid visiting pages or directories that have already been  
 378 visited through another path.

379 -- The referencing node retains its properties as child node, but the properties of the referenced  
 380 node in its role as parent remain intact. In the example in Figure 7, nodes E and G must use the  
 381 rollup rules defined by N for its children. If N references a RCD, and E also references a RCD,  
 382 the RCDs are considered equivalent. In this case, when E is inspected in the context of its par-  
 383 ent, i.e. when E is playing the role of child, the RCD that is visible is the RCD referenced by  
 384 E. However, when E is inspected in the context of its children, the RCD that is visible is the  
 385 RCD referenced by N, because E in its role as parent inherits the properties of N.

386 -- If the target of a reference is a DAG with more than one origin, the reference must specify  
 387 which origin to use. Otherwise it cannot be resolved. This adds some complication to imple-  
 388 mentations. For this reason, application profiles may want to restrict implementations to sin-  
 389 gle-origin DAG topologies.

## 390 **Using a DAG map to represent equivalencies and similarities**

391 A known and expected problem with the RCD model is that different communities may define  
 392 equivalent or similar RCDs because they are unaware of each other's work. This becomes a  
 393 problem when competency models or records must be reconciled. This can happen, for exam-  
 394 ple, when two companies merge or when educational Standards are consolidated across juris-  
 395 dictions. Since SRCMs can be used to represent relationships between RCDs, a natural use of  
 396 SRCMs is to represent simple equivalencies and similarities. For example, the SRCM that de-  
 397 fines such a competency equivalency can be referenced as evidence to justify the updating of a  
 398 competency record that references the equivalent competency.



399

400

**Figure 8 — A SRCM representing a one-way equivalency.**

401 Figure 8 shows how a straightforward equivalency can be represented in a SRCM. Figure 8  
 402 should be read as follows: Y is the only child of X, therefore proficiency in Y implies profi-  
 403 ciency in X. Note that the reverse equivalency, if it is true, cannot be represented in the same  
 404 SRCM because that would violate the acyclical constraint of a DAG. In practice, this is not a

405 problem, because competency equivalencies are often not reversible. For example, a country  
 406 may accept diplomas from another country, but the reverse may not be true without some ad-  
 407 ditional qualifications, as illustrated in Figure 9.

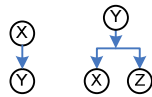
408 Figure 9 should be read as "Y has children X and Z. Therefore proficiency in Y requires profi-  
 409 ciency in X as well as proficiency in Z".



410

411 **Figure 9 — A SRCM representing a qualified reverse equivalency**

412 The equivalency assumptions between competencies X and Y can thus be expressed with two  
 413 maps, as shown in Figure 10. Figure 10 should be read as follows: "Two SRCMs represent the  
 414 relationships between competencies X and Y. Proficiency in Y implies proficiency in X. Pro-  
 415 ficiency in X implies proficiency in Y only if there is also proficiency in Z."



416

417 **Figure 10 —Set of SRCMs representing the equivalency between competencies X and Y**

#### 418 5.1.4 Data capture

419 The data in a SRCM can come from many sources. Typical existing sources include various  
 420 competency models developed by government agencies, academic institutions, enterprise and  
 421 military training departments.

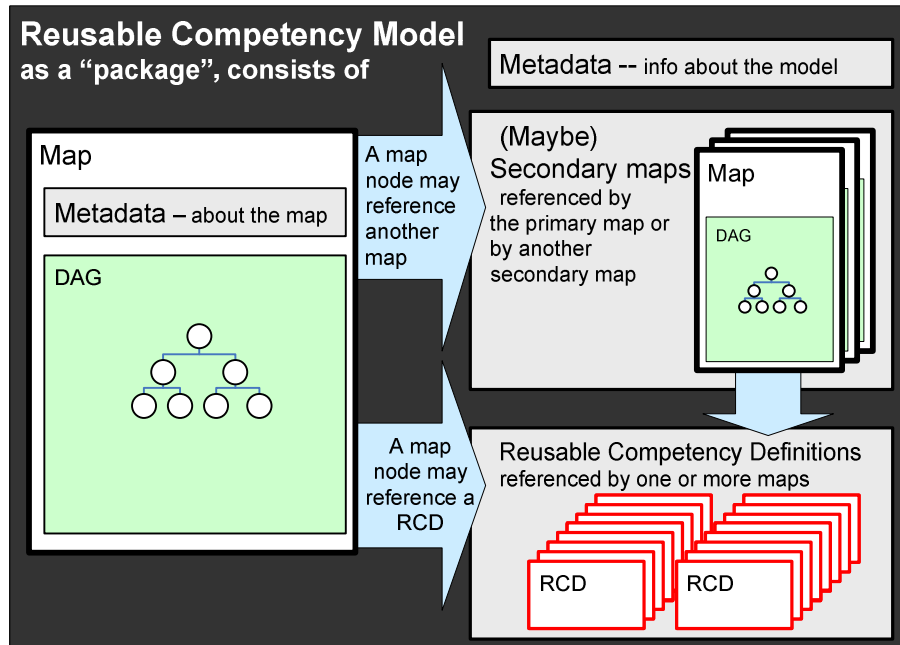
422 Many existing models are hierarchies or taxonomies that can be captured in more or less  
 423 automatic ways, depending on how they are encoded. The process might involve splitting the  
 424 data from the model into two data models – RCD (definitions) and SRCM (how the defini-  
 425 tions are related). A useful refinement on the process might be to look up existing RCD re-  
 426 positories to see whether suitable RCDs have already been defined, and then use those RCD  
 427 rather than creating new ones when possible.

428 The United States O\*NET database can be mined for specific competency maps correspond-  
 429 ing to a job description that someone builds by walking through the Standard occupational  
 430 descriptions and ability weights captured in O\*NET. For example, for a job that involves ac-  
 431 counting and management of a loading doc as well as supervision of twenty employees, an  
 432 HR person with the appropriate helper tool could identify the appropriate Standard occupation  
 433 codes in O\*NET and automatically generate a map of the required competencies and relative  
 434 importance for the required position, tweak it for the specific requirements of the enterprise,  
 435 then publish the map or pass it on to recruiting agencies.

436 Standards based technical documents such as S1000D conformant instances can also be mined  
 437 for the structure and content of a competency map. For example, a S1000D compliant techni-

438 cal manual for the maintenance and operation of a vehicle could be used as the source for  
 439 competency maps segmented for maintenance personnel and vehicle operators. In this exam-  
 440 ple, competency maps could also be updated automatically or semi-automatically as the de-  
 441 scribed equipment evolves. The application that extracts this information from the S1000D  
 442 instance could automatically guide the generation of new reusable competency definitions and  
 443 the creation or retirement of competency map nodes corresponding to feature additions or re-  
 444 movals in the vehicle's documentation.

### 445 5.1.5 Example applications



446

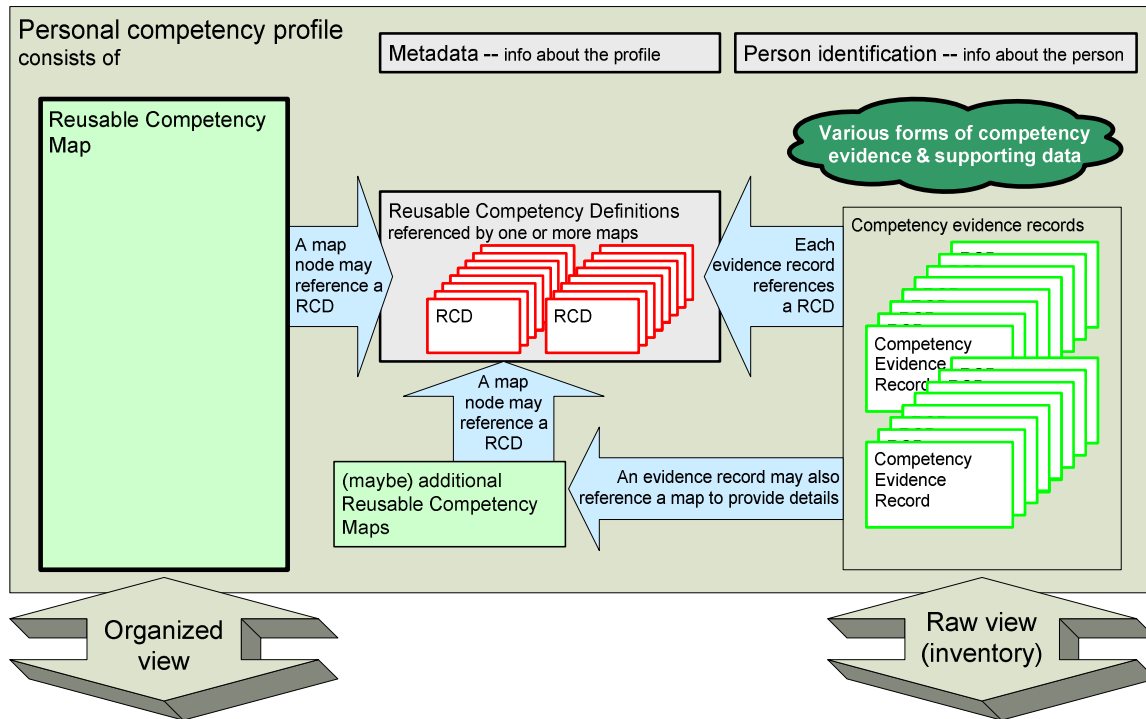
447 **Figure 11 — Application example: Packaging a job competency profile**

448 A competency map can be used to represent the relationships between competency definitions  
 449 in:

- 450 - a job competency profile (requirements for the job)
- 451 - a personal competency profile (acquired competencies)
- 452 - a personal competency gap profile (competencies to learn)

453 Some other existing or emerging competency data collection specifications or practices may  
 454 not flexible enough to represent all models, or do not support any meaningful interoperability.  
 455 This proposed map Standard, along with related Standards, would allow the capture of key  
 456 competency assumptions into a Standard, interoperable way. The competency information  
 457 could be captured by translating from proprietary or portfolio to a competency map with asso-  
 458 ciated RCDs and possibly competency evidence records if that is part of the original data.

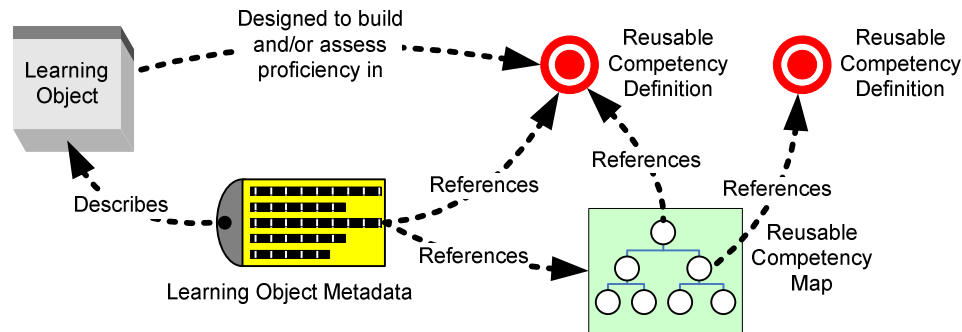
459 An application of this would be to allow look up of corresponding nodes in a map detailing  
 460 the specific sub-competencies and their relative importance for a job requirement, with the  
 461 RCD as a key (including the recording of equivalencies or dependencies discovered in ontolo-  
 462 gies, for example) Once a corresponding node is discovered, its relation links can be explored  
 463 to guide further analysis. This analysis does not have to be automated to be useful--it could be  
 464 presented as a guide to a human examiner. For example, if the match is on something like  
 465 "English verbal skills", the examiner could be presented with an outline of the sub-skills re-  
 466 quired for the job and compare that with the outline of sub-skills derived from the map that  
 467 results form the capture of the applicant's portfolio into a competency map.



468

469 **Figure 12— Application example: A personal competency profile**

470 Another example application would be in connection with learning object metadata, as shown  
 471 in Figure 13. The competency taught by a learning object, or tested by an assessment object,  
 472 should be represented by a reference to a reusable competency definition in the metadata. In  
 473 IEEE LOM, this would be defined in one or more Classification elements. Someone looking  
 474 for learning resources for a particular competency could then locate any competency map that  
 475 contains a reference to the corresponding reusable competency definition, and then use that  
 476 map to find higher level, related competency definitions. Searching for learning objects that  
 477 specify those reusable competency definitions may then find learning objects that contain  
 478 relevant resources, even when a one to one match on the original competency definition might  
 479 fail. This can be made more reliable if the competency model assumed by the designers of the  
 480 learning resource is made available in the metadata by referencing a reusable competency  
 481 map.



482

483

**Figure 13— Application example: Finding related learning resources**

## 484 5.2 Data model overview

485 The data model contains the following mandatory elements:

486 a) Identifier: A globally unique label that identifies this Reusable Competency Map. This  
 487 identifier uses the same data elements as the Identifier element defined in the IEEE LOM  
 488 Standard, and consists of two sub-elements: Catalog and Entry. The Identifier is sufficient to  
 489 reference the competency in any other system. The Identifier may be a handle or digital object  
 490 identifier according to emerging Standards and practice, e.g. the Handle system and the  
 491 CORDRA specification.

492 b) Title: A text label for the Reusable Competency Map. This is a short human-readable  
 493 name for the taxonomy. While the Identifier provides the definitive reference to the definition,  
 494 it is typically unintelligible. The Title provides a convenient alternative readable form, but one  
 495 which is not the definitive label. The Title may be repeated in multiple languages.

496 The other elements defined by the data model are optional:

497 c) Description: A human readable description of the map. This is an optional unstruc-  
 498 tured (opaque) “text blob” meant to be interpretable only by humans. The Description may be  
 499 repeated in multiple languages.

500 d) Other information: Other information about the map, including optional metadata.

501 d) Graph: A structured collection of nodes that represent the hierarchical relationship be-  
 502 tween competencies. There is always at least one node. Some or all the nodes may reference a  
 503 Reusable Competency Definition, which is not part of the data model, or another map that is  
 504 outside the scope of the map that contains the node. This reference is through an identifier  
 505 with global scope and does not specify a particular location, but rather the identifier of the  
 506 RCD or map, wherever it may be found. A node does not have to reference a RCD or map.  
 507 This may be because the node exists only for the purpose of grouping, because an appropriate  
 508 RCD or map has not yet been found or created for reference, or because a reference to a RCD  
 509 or map was found to be invalid and has been removed.

510 A node has a Title, which is a human readable name for the node. This Title may be repeated  
511 in multiple languages. If no Title is defined for a node, but the node references a Reusable  
512 Competency Definition or map, the Title may be obtained automatically from that Compe-  
513 tency Definition or map. An implementation may also obtain additional data from the refer-  
514 enced object by looking up the referenced object.

515 A node may have associated rules that specify how implementations may aggregate summary  
516 information about mastery of the referenced competencies. For example, one set of rules ap-  
517 plies when the node is considered as a parent in the hierarchy, and another set applies when  
518 the node is considered as a child in the hierarchy. In the absence of explicit rules specified by  
519 the creator of the map, implicit default rules are specified by this Standard.

520 This Standard does not define a specific extension mechanism for the data model. Implemen-  
521 ters may create additional data models for competency data and the representation or encoding  
522 of relationships between competencies. Such models may be used to augment this model to  
523 support different communities of practice.

524 (End of informative clauses)

## 525 **6. Data model**

526 This Clause defines the data elements of a SRCM

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

528 NOTES:

529 1—The use of ISO/IEC 11404 notation in the synopses in 6.1 and 6.2 is for descriptive purposes  
530 only. A complete implementation of the operations defined in ISO/IEC 11404 is not required for  
531 conformance.

532 2—The ISO/IEC 11404 notation describes the semantics of the language-independent data types  
533 across all bindings (e.g., implementation of a data type as itself, its subtypes, its subclasses, and  
534 its specializations). For example, an ISO/IEC 11404 "record" may be implemented as an SQL  
535 table row, or as an XML complexType; an ISO/IEC 11404 "characterstring" may be implemented  
536 in an encoding (ISO 646, ASCII, ISO 8859-1, UTF-8, UTF-16, UTF-32, etc.) that supports the  
537 repertoire specified in the parameter to characterstring data type.

538 3—All examples in 6.1 and 6.2 are informative and do not endorse any particular binding.

539 4—The following language-independent data types used in this Standard are defined in ISO/IEC  
540 11404: bag, characterstring, record, set, state.

541 5—The labels for data elements and data types in the synopses in clauses 6.1 and 6.2 are for ref-  
542 erence only. There is no requirement that an implementation use the exact same labels, as long as  
543 the data elements and data types are semantically equivalent.

544 6—This Standard does not define a specific extension mechanism for the data model. Implemen-  
 545 ters may define binding that allow additional elements, or create additional data models for com-  
 546 petency data. Such models may be used to augment this model to support different communities  
 547 of practice.

## 548 6.1 Reusable Competency Map

### 549 Synopsis

```
550 reusable_competency_map :
551 record
552 (
553   identifier : long_identifier_type,
554   rcdRef : long_identifier_type,
555           // optional
556   title : langstring_type(1000),
557           // the parameter value is the SPM
558           // optional
559   description : langstring_type(4000),
560           // the parameter value is the SPM
561           // optional
562   SRCM_schema_label: characterstring(iso-10646-1),
563           // SPM: 1000 characters
564   SRCM_schema_version: characterstring(iso-10646-1),
565           // SPM: 1000 characters
566   referential: boolean,   metadata : bag of any_type,
567           // SPM: 10 of any type in the bag
568   graph : ( see clause 6.1.8),
569   extensions : any_type
570 )
```

### 571 Description

572 The components of `reusable_competency_map` are defined in 6.1.1 – 6.1.9. Depending  
 573 on the implementation, an instance of `reusable_competency_map` shall include one or  
 574 more of the defined components. The map element is a required component. It may not be  
 575 empty.

576 All top level elements in the `reusable_competency_map` data model are intrinsically un-  
 577 ordered.

578 Note—A binding may impose a particular ordering on SRCM data elements that conform to that  
 579 binding. Other than conformance to the binding, no significance is associated or should be in-  
 580 ferred from that ordering requirement.

## 581 **6.1.1 Identifier**

### 582 **Synopsis**

```
583     identifier :  
584         long_identifier_type,
```

### 585 **Description**

586 A globally unique label that identifies this Reusable Competency Map. The Identifier is suffi-  
587 cient to reference the competency taxonomy in any other system.

588 Subclause 6.2.1 defines long\_identifier\_type.

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

## 591 **6.1.2 Title**

### 592 **Synopsis**

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

### 596 **Description**

597 A single mandatory text label for this SRCM. This is a short human-readable name for the tax-  
598 onomy.

599 Subclause 6.3.1 defines langstring\_type.

600 NOTES:

601 1—The Title may be repeated in multiple languages.

602 2— While the Identifier provides the definitive reference to the SRCM, it is typically unintelligi-  
603 ble. The Title provides a convenient alternative readable form, but one which is not the definitive  
604 label. Examples: "English proficiency", "Schmiblick failure diagnostic level 4", "Demonstrates  
605 conflict resolution skills".

## 606 **6.1.3 Description**

### 607 **Synopsis**

```
608     description : bag of langstring_type(4000),  
609         // SPM: 20 instance of langstring_type in the bag  
610         // The parameter value is the SPM for the langstring
```

**611 Description**

612 A human readable description of the competency map. This is an optional unstructured char-  
613 acter string meant to be interpretable only by humans or unstructured full text indexing  
614 schemes.

615 Subclause 6.2.1 defines langstring\_type.

616 NOTES:

617 1—The Description may be repeated in multiple languages.

618 2—The description is typically more explicative than the title. Examples: "Proficiency in written  
619 and spoken English and use of English for meaningful oral or written expression.", "Performance  
620 of level 4 diagnostic as specified in IETM #SCMBLK007"

**621 6.1.4 SRCM schema label****622 Synopsis**

```
623     SRCM_schema_label : characterstring(iso-10646-1),
624         // SPM: 1000 characters
```

**625 Description**

626 A label for the schema that defines and controls this SRCM data.instance. If this element is  
627 omitted then a value of “IEEE 1484.20??? (td)” shall be assumed. A conforming implemen-  
628 tation may refuse to accept or process a SRCM instance if the value of this data element is  
629 specified, but is not the value specified in the Standard.

630 NOTES

631 1—Different label values may be used to signal application profiles with particular extensions or  
632 processing requirements.

633 2—This element does not describe the schema of the embedded metadata defined in 6.1.7. Every  
634 instance of embedded metadata, if any, should include its own schema description.

**635 6.1.5 SRCM schema version****636 Synopsis**

```
637     SRCM_schema_version : characterstring(iso-10646-1),
638         // SPM: 1000 characters
```

**639 Description**

640 Describes the version of the schema described by SRCM\_schema. If this element is omitted  
641 then a value of “1.0” shall be assumed. A conforming implementation may refuse to accept or

642 process a SRCM instance if the value of this data element is specified, but is not the value  
643 specified in the Standard.

644 NOTE—1—Different values may be used to signal application profiles with particular extensions  
645 or processing requirements.

## 646 **6.1.6 Referential**

### 647 **Synopsis**

648 `referential : boolean,`

### 649 **Description**

650 The `referential` data element indicates whether the map is self-contained or includes ref-  
651 erences to other maps. If the map contains references to other maps, this element shall be re-  
652 quired and its value shall be true. This element shall be optional if the map does not contain  
653 references to other maps, in which case its default value shall be assumed to be false.

654 Note. The entity that creates the map is responsible for setting the value of this element to ac-  
655 curately represent whether the map includes references to other maps.

656 Note 2: This element is included to facilitate processing of maps, e.g. to help determine  
657 whether a map has external dependencies without having to examine every node of the map.

## 658 **6.1.7 Metadata**

### 659 **Synopsis**

660 `metadata :`  
661 `bag of any_type, // SPM: 10 of any type in the bag`

### 662 **Description**

663 Optional embedded Metadata describing this SRCM.

664 If a metadata record is included, it is recommended that this record conform to IEEE 1484-  
665 12.1-2002: Standard for Learning Object Metadata (IEEE LOM). In such conforming records,  
666 the version of the Metadata Specification Standard is given in the meta-metadata element of  
667 the metadata record. Some of the data elements defined in IEEE LOM are not relevant for a  
668 SRCM and should be omitted. Profiles shall determine which metadata, if any, must be used  
669 for conformance with the profile.

670 NOTES:

671 1—Useful metadata defined in the IEEE LOM include additional identification as an entry in  
672 one or more catalogues, information about the author, publisher, the creation date, and the  
673 coverage (in the sense of the Dublin Core as adopted by the IEEE LOM.) The Relation ele-  
674 ment may be used to relate a definition to a prior version of the definition, and one or more

675 Classification elements may be used to indicate where this particular map fits in a larger  
 676 SRCM, map, model or ontology of competencies or educational objectives. Classification  
 677 elements may be repeated to reference different models. For example, the SRCM might repre-  
 678 sent a map of a competency as defined in O\*NET model, as well as a map of a similar compe-  
 679 tency defined in a company's custom competency model.

680 2—More than one metadata record is allowed in the bag, but if there is more than one record  
 681 each record should conform to a different metadata specification. An implementation must  
 682 accept any metadata record that it cannot interpret, but it is not required to interpret such  
 683 metadata records.

684 3—A particular binding specification or application profile may impose additional restrictions  
 685 or requirements.

## 686 **6.1.8 Graph**

### 687 **Synopsis**

```
688 graph : record :
689 (
690     default_entry_node: local_identifier_type,
691     entry_nodes: bag of local_identifier_type,
692     nodes : bag of node_type,
693     // SPM: (To be discussed. Specify in profiles?)
694 )
```

### 695 **Description**

696 A mandatory element. This element contains the actual map in the form of a collection of di-  
 697 rected graph nodes. Any node without a parent is an entry node into the graph.

698 NOTES:

699 1— A binding might require a particular ordering of the elements. Such ordering does not imply  
 700 any meaningful order.

701 2— Some competency models prescribe an ordering or precedence for competencies. This typi-  
 702 cally means that competencies must be mastered in a specific order. This, in turn, means that  
 703 there are dependencies between the competencies so that mastery in A is a requirement for mas-  
 704 tery in B, B is a requirement for mastery of C, and so on. This can be described in the DAG by  
 705 making A a child of B, and B a child of C, etc. without adding an ordering property.

### 706 **6.1.8.1 Default entry node**

#### 707 **Synopsis**

708 `default_entry_node: local_identifier_type,`

#### 709 **Description**

710 This element specifies the default entry node into the graph. It is optional, but an application  
711 profile may require it to support applications that require a single entry node even if the graph  
712 contains more than one entry node.

### 713 **6.1.8.2 Entry nodes**

#### 714 **Synopsis**

715 `entry_nodes: bag of local_identifier_type,`

#### 716 **Description**

717 This element contains a complete and exact list of the nodes that do not have at least one par-  
718 ent. There is no intrinsic order to the list.

719 Note – This element is provided to accelerate processing by not requiring a traversal of the  
720 entire node collection to identify the entry nodes.

### 721 **6.1.8.3 Nodes**

#### 722 **Synopsis**

723 `nodes : bag of node_type :`

#### 724 **Description**

725 Unordered collection that contains all the nodes that make up the DAG. There is no intrinsic  
726 order to the nodes, other than the order defined by parent-child relationships. Node\_type is  
727 defined in clause 6.2.1

### 728 **6.1.9 Extensions**

#### 729 **Synopsis**

730 `extensions : any_type`

#### 731 **Description**

732 Application specific extensions to the data model. The extensions may not conflict with a  
733 binding of the data model. Extensions may not replace or duplicate elements defined in the  
734 data model.

735 NOTES:

736 1—The extensions should be defined by an application profile.

## 737 6.2 Data types

738 The following data types are used in conjunction with the data elements described in Clauses  
739 6.1 and 6.2.

### 740 6.2.1 Node type

#### 741 Synopsis

```

742     node_type = record :
743     (
744         identifier :
745             local_identifier_type,
746             // SPM: 1000 characters
747         rcd_reference :
748             long_identifier_type,
749             // SPM: 4000 characters;
750         title :
751             bag of langstring_type(1000),
752             // SPM: 20 instance of langstring_type in the bag
753             // The parameter value is the SPM for the
754             // langstring
755         description :
756             bag of langstring_type(2000),
757             // SPM: 20 instance of langstring_type in the bag
758             // The parameter value is the SPM for the
759             // langstring
760         parents :
761             bag of long_identifier_type,
762             // nil if no parent
763         children :
764             bag of long_identifier_type,
765             // nil or empty if no children
766             // must be nil if sym_link is not nil
767         sym_link : long_identifier_type,
768             // must be nil if children is not nil
769         rules :
770             // see clause ....
771     ),
```

#### 772 Description

773 This type defines the data model for a single node in the DAG that represents the map's topol-  
774 ogy. The following clauses describe the elements within each node record.

### 775 **6.2.1.1 Identifier**

#### 776 **Synopsis**

```
777     identifier : local_identifier,
```

#### 778 **Description**

779 A unique label that identifies the node. The Identifier is sufficient to reference the node in the  
780 graph. This Identifier shall be unique at least in the context of the SRCM.

781 Subclause 6.x.xxx defines long\_identifier\_type.

#### 782 **NOTES:**

783 1—If there is any possibility that the map may be disaggregated or merged with other maps or  
784 fragments of maps, it is recommended that the Identifier be globally unique. The result of merg-  
785 ing maps that contain nodes with conflicting identifiers is undefined.

### 786 **6.2.1.2 RCD Reference**

#### 787 **Synopsis**

```
788     rcd_reference :  
789         long_identifier_type,  
790         // SPM: 4000 characters
```

#### 791 **Description**

792 The identifier of the Reusable Competency Definition represented by this node. If the node  
793 exists only for the purpose of grouping other nodes in the map, this element may be omitted or  
794 its value may be nil.

### 795 **6.2.1.3 Title**

#### 796 **Synopsis**

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

#### 800 **Description**

801 A short human readable text label for the node.

802 If the node references a Reusable Competency Definition and this element has a value, im-  
803 plementations shall display the title as defined by this element instead of the title of the refer-  
804 enced Reusable Competency Definition. If the node references a Reusable Competency Defi-  
805 nition and this element has no value, implementations should display the title as defined by the  
806 referenced Reusable Competency Definition. However, this is not a requirement because per-

807 formance or other considerations may make this impractical. If no title is available for display,  
808 but it is necessary to display the node, the implementation shall define what to display.

#### 809 **6.2.1.4 Description**

##### 810 **Synopsis**

```
811     description : bag of langstring_type(2000),  
812         // SPM: 20 instance of langstring_type in the bag  
813         // The parameter value is the SPM for the langstring
```

##### 814 **Description**

815 A human readable text description for the node.

816 If the node references a Reusable Competency Definition and this element has a value, im-  
817 plementations shall display the description as defined by this element instead of the descrip-  
818 tion in the referenced Reusable Competency Definition. If the node references a Reusable  
819 Competency Definition and this element has no value, implementations should display the  
820 description as defined by the referenced Reusable Competency Definition, if available. How-  
821 ever, this is not a requirement because performance or other considerations may make this im-  
822 practical.

#### 823 **6.2.1.5 Parents**

##### 824 **Synopsis**

```
825     parents : bag of long_identifier_type,
```

##### 826 **Description**

827 Identifiers of the parent nodes of this node in the directed graph topology. This element shall  
828 be null, empty or omitted if the node does not have any parent.

829 NOTE— In a DAG, a node may have more than one parent. A corollary is that several nodes may  
830 share one or more children.

#### 831 **6.2.1.6 Children**

##### 832 **Synopsis**

```
833     children : bag of long_identifier_type,
```

##### 834 **Description**

835 Identifiers of the child nodes of this node in the directed graph topology. This element shall be  
836 null, empty or omitted if the node has no children. A node shall be prohibited to have children  
837 if a sym\_link to a different map is specified for the node.

838 **6.2.1.7 Sym Link**839 **Synopsis**840 `sym_link : long_identifier_type,`841 **Description**

842 Identifier referencing a map used to extend the current map by merging an entry node in that  
 843 map with the current node. The binding of the identifier may include the specification of a par-  
 844 ticular entry node in the target map. If the target map has multiple entry nodes and no entry  
 845 node is specified in this element, the default entry node in the target map shall be used as entry  
 846 node. The merging rules are specified in clause x.x.

847 **NOTES:**

848 1— See conceptual overview for a detailed informative explanation of the map extension mecha-  
 849 nism.

850 2— By using sym links, it is possible to keep SRCMs to an easily manageable size, and to reuse  
 851 SRCMs in the context of other maps. For example, the same skill definition published by a state  
 852 standards board may included in institution-specific competency models through this mechanism.

853 **6.2.1.8 Rules**854 **Synopsis**

```
855 rules : record
856 (
857     // optional rules to support applications
858     proficiency_threshold: real(10,7), // range (0..1),
859     // rules that apply to node in parent role
860     rollup_method : state(all,any,fraction,units,mean,other),
861     rollup_parameter : choice(rollup_method) of
862     (
863         all, any, mean : nil,
864         fraction: real(10,7), // range (0..1),
865         units: integer, // must be > 0
866         other: characterstring(iso-10646-1),
867         // SPM: 4000
868     )
869     // rules that apply to node in child role
870     as_child_weight: real(10,7), // range (0..1),
871     as_child_required: boolean, // default = true;
872 )
```

873 **Description**

874 Rules that specify the data processing behavior of systems that use the SRCM. There are three  
 875 groups of rules. Application support rules are used to specify threshold values useful in appli-  
 876 cations. Rules as parent are used only when the node is used as a parent, to specify how to roll

877 up proficiency information for competencies referenced by child nodes. Rules as child are  
878 used only when the node is a child in a rollup process.

879 If the rules element is not present in an instance, the default value for each the rule elements  
880 defined here shall be applied in processing rules. If any of the rule elements is not specified,  
881 the default value for that element shall be assumed in processing rules.

### 882 **6.2.1.9 Proficiency threshold**

#### 883 **Synopsis**

884 `proficiency_threshold: real(10,7), // range (-1..1),`

#### 885 **Description**

886 This data item specifies whether proficiency status can be assumed to be true if the available  
887 proficiency measure for the referenced competency definition is at least a certain value. For  
888 valid comparisons, the proficiency measure must be expressed in a compatible range. The  
889 range for `proficiency_threshold` is -1 to 1 inclusive. The default default value if not  
890 explicitly specified is 1.

#### 891 **NOTES:**

892 1— Proficiency measure is also called "proficiency level", "score" or "success measure" in vari-  
893 ous specifications and Standards.

894 2— Application example: The current node references RCD X and a competency evidence record  
895 for an individual also points to RCD X, and specifies that the individual's proficiency level is  
896 67%. If a proficiency threshold of 70% (=0.7) is specified by for this node, the individual is not  
897 considered proficient according to this competency map.

### 898 **6.2.1.10 Rollup method**

#### 899 **Synopsis**

900 `rollup_method : state(all,any,fraction,units,mean,other),`

#### 901 **Description**

902 This data item applies only when the node is a parent. It specifies how to roll up proficiency  
903 information specified by children of the node. The defined methods are:

- 904 • `all` -- Proficiency must be achieved for all the competencies represented by chil-  
905 dren nodes in order to consider that proficiency is achieved for the competency repre-  
906 sented by this node. This is the assumed default value if a rollup method is not speci-  
907 fied.
- 908 • `any` -- Proficiency must be achieved for any of the competencies represented by  
909 children nodes in order to consider that proficiency is achieved for the competency  
910 represented by this node.

- 911 • fraction -- Proficiency must be achieved for at least specified fraction of the compe-  
912 tencies represented by children nodes in order to consider that proficiency is achieved  
913 for the competency represented by this node. The fraction value is expressed as a  
914 floating point number in the range 0 to 1 inclusive. 0 means that no child proficiency is  
915 required. 1 is the equivalent of specifying that the rollup method is all. 0.5 means that  
916 proficiency must be achieved for at least 50% of the child nodes.
- 917 • units -- Proficiency must be achieved for at least the specified number of the compe-  
918 tencies represented by children nodes in order to consider that proficiency is achieved  
919 for the competency represented by this node. The number is a positive integer. 0  
920 means that no child proficiency is required. 1 is the equivalent of specifying that the  
921 rollup method is any.
- 922 • mean -- The proficiency measure for the parent node is determined by averaging the  
923 proficiency levels for the competencies represented by children nodes. The proficiency  
924 status for the parent node can then be determined by comparing that level with the pro-  
925 ficiency\_threshold value for the parent node. If no proficiency level is available for a  
926 child node, the proficiency status (boolean) is used instead, with proficient = 1 and not  
927 proficient or unknown = 0.
- 928 • other -- Another proficiency method is to be used. The other method must be speci-  
929 fied by an application profile and is not defined by the Standard.

930 By default, the rollup method is "all". No available competency data or a value of "unknown"  
931 is considered the same as "not proficient" for the purpose of rollup. If proficiency status but no  
932 proficiency measure is available for any node, the proficiency measure value is assumed to be  
933 1 if proficient, and 0 if not proficient.

934 NOTES:

935 1— Proficiency measure is also called "proficiency level", "score" or "success measure" in vari-  
936 ous specifications and Standards.

### 937 **6.2.1.11 Rollup parameter**

#### 938 **Synopsis**

```
939 rollup_parameter : choice(rollup_method) of
940   (
941     all, any, mean : nil,
942     fraction: real(10,7), // range (0..1),
943     units: integer, //
944     other: characterstring(iso-10646-1),
945     // SPM: 4000
```

946 )

#### 947 **Description**

948 Parameter used only if the specified rollup method is fraction, units or other. The type of pa-  
949 rameter depends on the rollup method.

950 If the rollup method is "other", the parameter is a container for data of any type, and an appli-  
951 cation profile must specify the method and the data.

#### 952 **6.2.1.12 Weight as child**

##### 953 **Synopsis**

```
954     as_child_weight: real(10,7), // range (0..1),
```

##### 955 **Description**

956 This data element is used only when the node is considered as a child in a rollup operation  
957 where a weight assigned to the information associated with the child node is relevant. For ex-  
958 ample, if the rollup method is mean, this data element can be used to specify a relative weight  
959 among children. In the absence of a specific value, the default value of this element is assumed  
960 to be 1.

#### 961 **6.2.1.13 Required as child**

##### 962 **Synopsis**

```
963     as_child_required: boolean, // default = true;
```

##### 964 **Description**

965 This data element is used only when the node is considered as a child in a rollup operation. If  
966 its value is false, then the child is included in rollups only if valid proficiency information is  
967 available for it. The child is ignored altogether if no valid proficiency information is available  
968 for it. In the absence of a specific value, the default value of this element is assumed to be true.

969 Note – This element allows certain rollup operations where proficiency information should be  
970 used if available, but ignored if not available. For example, an individual may be considered a  
971 good player if she is competent in all the sports for which proficiency data are available for  
972 her, even though the model may list additional sports in which she never participated.

## 973 6.2.1 Any type

### 974 Synopsis

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

### 976 Description

977 This data type represents any type not specified in this Standard. This Standard does not re-  
 978 quire an implementation to process data elements of this type when encountered in a RCD  
 979 instance. The data types implemented in a SRCM data element with type any\_type shall not  
 980 preclude interoperable bindings of the entire SRCM instance.

### 981 NOTES

982 1— If implementations specify or require data elements for which the type is defined in this Stan-  
 983 dard as any\_type, this Standard recommends that the implementations provide the means to inter-  
 984 pret and validate the implementation-specific data. For example, an implementation that uses an  
 985 XML binding should include a valid XML schema that can be referenced in RCD instances  
 986 bound in XML documents, and the schema should be documented.

987 2—The data types used in the container should be defined by an application profile.

### 988 Examples

989 (tbd)

## 990 6.2.2 LangString type

### 991 Synopsis

```
992     type langstring_type(length) =
993         record
994             (
995                 language :
996                     language_type,
997                 string :
998                     characterstring(iso-10646-1)
999                     // SPM: The length parameter
1000             );
```

### 1001 Description

1002 This data type consists of a language specification for a string and the string itself.

### 1003 Examples

1004 The following are three examples of localized strings: "Information Technology" in  
 1005 French, "localization" in British English, and "xxx" in Japanese hiragana.

```

1006     ( "fr", "Technologies de l'information" ),
1007     ( "en-GB", "localisation" ),
1008     ( "jp-JP-jisx208", "xxx" ),

```

### 1009 6.2.2.1 Language

#### 1010 Synopsis

```

1011     language :
1012         characterstring(iso-646),
1013         // SPM: 250 characters

```

#### 1014 Description

1015 The `language` data element specifies the language of the localized string. The format of this  
 1016 data type is a character string consisting of a required language code followed by multiple,  
 1017 optional, hyphen-prefixed subcodes (see examples below).

1018 The following rules apply to the language code part of the character string :

- 1019 – 2-letter codes are defined by ISO 639–1.
- 1020 – 3-letter codes are defined by ISO 639–2.
- 1021 – The value prefix "i" is reserved for registrations defined by the Internet As-  
 1022 signed Numbers Authority (IANA).
- 1023 – The value prefix "x" is reserved for private use.

1024 The following rules apply to the first subcode part of the character string :

- 1025 – 2-letter subcodes are ISO 3166–1 alpha-2 country codes.
- 1026 – Subcodes of from 3 to 8 letters are registered with IANA.

1027 Rules for additional subcodes are unspecified, except that the length of any subcode cannot  
 1028 exceed 8 characters.

1029 NOTE—The language code is often given in lower case and the subcodes (if any) in upper case.  
 1030 However, the values are case insensitive.

#### 1031 Examples

```

1032     "en-GB"
1033     "de"
1034     "fr-CA"
1035     "it"
1036     "grc" (Ancient Greek, until 1453)
1037     "en-US-NY-NY"
1038     "eng-GB-cockney"
1039     "map-PG-buin" (Austronesian - Papua New Guinea Buin)
1040     "gem-US-PA"

```

### 1041 **6.2.3.2 String**

#### 1042 **Synopsis**

```
1043     string :  
1044         characterstring(iso-10646-1),  
1045         // SPM: The length parameter
```

#### 1046 **Description**

1047 The `string` data element contains the text of the localized string.

### 1048 **6.2.3 Long identifier type**

#### 1049 **Synopsis**

```
1050     type long_identifier_type =  
1051         record (  
1052             catalog: characterstring(iso-10646-1),  
1053             // SPM: 4000 characters  
1054             entry: characterstring(iso-10646-1)  
1055             // SPM: 4000 characters  
1056         );
```

#### 1057 **Description**

1058 This data type is an identifier (a label) associated with an object that is intended to be unique  
1059 within the context of usage of the object. The character string shall conform to the syntax for  
1060 Uniform Resource Identifiers (URIs) as defined by RFC 2396.

1061 NOTE—This Standard recommends that the URI be a globally unique identifier in the form of a  
1062 Uniform Resource Name (URN) (see RFC 2141 [A5]) or a Handle as defined in the Handle Sys-  
1063 tem [Ax].

### 1064 **6.2.3 Short identifier type**

#### 1065 **Synopsis**

```
1066     type short_identifier_type = characterstring(iso-10646-1)  
1067         // SPM: 4000 characters
```

#### 1068 **Description**

1069 This data type is an identifier (a label) associated with an object that is intended to be unique  
1070 within the context of usage of the object. The character string shall conform to the syntax for  
1071 Uniform Resource Identifiers (URIs) as defined by RFC 2396.

1072 NOTES

1073 1—This Standard recommends that the URI be a globally unique identifier in the form of a Uni-  
1074 form Resource Name (URN) (see RFC 2141 [A5]) or a Handle as defined in the Handle System  
1075 [Ax].

1076 2— Depending on the binding, these data elements may be implemented as explicit or implicit.  
1077 The binding may be a combination of explicit and implicit representations of these data elements.  
1078 For example, in an XML document they may be implicit in the structure of the document if the  
1079 shape of the graph is a simple tree.

1080 3— Depending on the implementation, the values in the parents and children collections may  
1081 be implicit, may be explicit and literal or may be an implementation-specific identifier or  
1082 pointer. However, any conforming implementation of a Reusable Competency Map must rep-  
1083 resent no less than the information specified in this Standard for any component that is present  
1084 in the implementation. For example, even it is a node is represented by a relative pointer to  
1085 achieve data compression in a particular binding, the identifier for that node must be resolv-  
1086 able to a conformant identifier type when the tree instance is transmitted to another confor-  
1087 mant implementation.

1088

1089

1090 **Annex B**

1091 [A2] IEEE 1484.11.2–2003, Standard for Learning Technology—ECMAScript Application  
1092 Programming Interface for Content to Runtime Services Communication.

1093 [A3] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

1094 [A5] RFC 2141, "URN Syntax," Network Working Group, May 1997.

1095 **IMSRDCEO** IMS Global Learning Consortium, Inc. (2002). IMS Reusable Defini-  
1096 tion of Competency or Educational Objective, Version 1.0

1097 **O\*NET** <http://online.onetcenter.org/> (or  
1098 [http://www.access.gpo.gov/o\\_net/datadict/datadict.pdf](http://www.access.gpo.gov/o_net/datadict/datadict.pdf))

1099 **Ostyn** <http://ltsc.ieee.org/doc/wg20/CompDefInit.doc> (Base document from  
1100 P1484.20)

1101 **PASS** <http://www.ous.edu/pass/Standards/admission.html> (Oregon Profi-  
1102 ciency-based Admissions Standards System)

1103 **SCANS** <http://www.tier.net/tcenters/scans.htm> (Secretary's Commission on  
1104 Achieving Necessary Skills: Competencies)

1105 **SCORM** <http://www.adlnet.org> (ADL SCORM)

1106 **TATS** <http://www.adtdl.army.mil/atdls.htm> (Total Army Training System)

1107 **Annex B**

1108 (informative)

## 1109 Sample XML Binding

1110 This sample shows an XML instance and the corresponding schema for an XML binding.  
 1111 This is a sample only. The instance shows that RCDs are known to exist for 3 competencies  
 1112 defined by different agencies: "Can drive a car safely", "Can steer a car and control its speed"  
 1113 and "Knows and understands road signs and markings", and that the first competency is com-  
 1114 posed of the other two. No rules are specified because the default rules are good enough.

```
1115 <?xml version="1.0" encoding="UTF-8"?>
1116 <simpleCompetencyMap xmlns="ltsc.ieee.org/study.20.tbd.draft/simpleCompetencyMap"
1117 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1118 xsi:schemaLocation="ltsc.ieee.org/study.20.tbd.draft/simpleCompetencyMap
1119 D:\Claude\Standards\IEEELT~1\RCD\Proposals\simple-competency-map-draft.xsd">
1120   <identifier>123356/123.45tyu/345xyz</identifier>
1121   <schemaLabel>SRCM0.1.0</schemaLabel>
1122   <schemaVersion>1.0</schemaVersion>
1123   <graph>
1124     <defaultEntryNode>someIdOrOther6</defaultEntryNode>
1125     <entryNode>someIdOrOther6</entryNode>
1126     <node>
1127       <identifier>someIdOrOther6</identifier>
1128       <rcdRef>http://somalicensingdept.gov/ref/driver/minimal</rcdRef>
1129       <title lang="en">Can drive a car safely</title>
1130       <children>
1131         <nodeId>someIdOrOther7</nodeId>
1132         <nodeId>someIdOrOther8</nodeId>
1133       </children>
1134     </node>
1135     <node>
1136       <identifier>someIdOrOther7</identifier>
1137       <rcdRef>http://somalicensingdept.gov/ref/driver/skl</rcdRef>
1138       <title lang="en">Can steer a car and control its speed</title>
1139       <parents>
1140         <nodeId>someIdOrOther6</nodeId>
1141       </parents>
1142     </node>
1143     <node>
1144       <identifier>someIdOrOther8</identifier>
1145       <rcdRef>http://usdot.gov/rcdsroadsigns2005</rcdRef>
1146       <title lang="en">Knows and understands road signs and markings</title>
1147       <parents>
1148         <nodeId>someIdOrOther6</nodeId>
1149       </parents>
1150     </node>
1151   </graph>
1152 </simpleCompetencyMap>
```

1153

```
1154 <?xml version="1.0" encoding="UTF-8"?>
1155 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1156 xmlns="ltsc.ieee.org/study.20.tbd.draft/simpleCompetencyMap"
1157 targetNamespace="ltsc.ieee.org/study.20.tbd.draft/simpleCompetencyMap"
1158 elementFormDefault="qualified" attributeFormDefault="unqualified">
1159   <xs:element name="simpleCompetencyMap" type="competencyDAGGraphType">
1160     <xs:annotation>
1161       <xs:documentation>
1162         Copyright 2005 Claude Ostyn - All rights reserved.
1163         Schema for a simple reusable competency map.
1164         A competency map may reference a Reusable Competency Definition
1165         or be standalone.
1166         The map represents a hierarchy of Competency Definitions.
1167         The nodes in the map are organized as a Directed Acyclic Graph (DAG)
1168         See proposed standard text for details.
1169         This is a strawman schema and should not be used for conformance.
1170       </xs:documentation>
```

```

1171     </xs:annotation>
1172 </xs:element>
1173 <!-- -->
1174 <!-- Overall type definition -->
1175 <!-- -->
1176 <xs:complexType name="competencyDAGraphType">
1177   <xs:sequence>
1178     <xs:element name="identifier" type="xs:anyURI" nillable="false"/>
1179     <xs:element name="rcdRef" type="xs:anyURI" minOccurs="0"/>
1180     <xs:element name="title" type="langString" minOccurs="0" maxOccurs="unbounded"/>
1181     <xs:element name="description" type="langString" minOccurs="0"
1182 maxOccurs="unbounded"/>
1183     <xs:element name="schema" type="xs:string" default="1.0"/>
1184     <xs:element name="schemaVersion" type="xs:string" default="1.0"/>
1185     <xs:element name="referential" type="xs:boolean" minOccurs="0" default="false"/>
1186     <xs:element name="metadata" type="metadataType" minOccurs="0"/>
1187     <xs:element name="graph" type="graphType"/>
1188     <xs:group ref="grp.any"/>
1189   </xs:sequence>
1190   <xs:anyAttribute namespace="##other" processContents="strict"/>
1191 </xs:complexType>
1192 <!-- -->
1193 <!-- Main sub type definitions -->
1194 <!-- -->
1195 <xs:complexType name="graphType">
1196   <xs:sequence>
1197     <xs:element name="defaultEntryNode" type="localNodeIdType"/>
1198     <xs:element name="entryNode" type="localNodeIdType" maxOccurs="unbounded"/>
1199     <xs:element name="node" type="nodeType" maxOccurs="unbounded"/>
1200     <xs:group ref="grp.any"/>
1201   </xs:sequence>
1202   <xs:anyAttribute namespace="##other" processContents="strict"/>
1203 </xs:complexType>
1204 <xs:complexType name="nodeType">
1205   <xs:sequence>
1206     <xs:element name="identifier" type="localNodeIdType" nillable="false"/>
1207     <xs:element name="rcdRef" type="xs:anyURI" minOccurs="0"/>
1208     <xs:element name="title" type="langString" minOccurs="0" maxOccurs="unbounded"/>
1209     <xs:element name="description" type="langString" minOccurs="0"
1210 maxOccurs="unbounded"/>
1211     <xs:element name="parents" type="relatedNodesType" minOccurs="0" />
1212     <xs:element name="children" type="relatedNodesType" minOccurs="0" />
1213     <xs:element name="symLink" type="xs:anyURI" minOccurs="0"/>
1214     <xs:element name="rules" type="rulesType" minOccurs="0"/>
1215     <xs:group ref="grp.any"/>
1216   </xs:sequence>
1217   <xs:anyAttribute namespace="##other" processContents="strict"/>
1218 </xs:complexType>
1219 <xs:complexType name="relatedNodesType">
1220   <xs:sequence>
1221     <xs:element name="nodeId" type="localNodeIdType" minOccurs="0"
1222 maxOccurs="unbounded"/>
1223     <xs:group ref="grp.any"/>
1224   </xs:sequence>
1225   <xs:anyAttribute namespace="##other" processContents="strict"/>
1226 </xs:complexType>
1227 <!-- -->
1228 <!-- Rules -->
1229 <!-- -->
1230 <xs:complexType name="rulesType">
1231   <xs:sequence>
1232     <xs:element name="proficiencyThreshold" type="xs:float" default="1"
1233 minOccurs="0"/>
1234     <xs:element name="rollupMethod" type="rollupMethodType" default="all"
1235 minOccurs="0"/>
1236     <xs:element name="rollupParam" type="xs:string" minOccurs="0"/>
1237     <xs:element name="asChildWeight" type="xs:float" default="1" minOccurs="0"/>
1238     <xs:element name="asChildRequired" type="xs:boolean" default="true"
1239 minOccurs="0"/>
1240     <xs:group ref="grp.any"/>
1241   </xs:sequence>

```

```

1242     <xs:anyAttribute namespace="##other" processContents="strict"/>
1243 </xs:complexType>
1244 <!-- -->
1245 <!-- Generic type definitions usable in various schemas -->
1246 <!-- -->
1247 <xs:complexType name="metadataType" abstract="false">
1248   <xs:annotation>
1249     <xs:documentation xml:lang="en">
1250       Placeholder for zero or more metadata instances of any type.
1251     </xs:documentation>
1252   </xs:annotation>
1253   <xs:sequence>
1254     <xs:group ref="grp.any"/>
1255   </xs:sequence>
1256   <xs:anyAttribute namespace="##other" processContents="strict"/>
1257 </xs:complexType>
1258 <xs:complexType name="localNodeIdType" abstract="false">
1259   <xs:annotation>
1260     <xs:documentation xml:lang="en">
1261       Identifier type for a node contained within this XML instance.
1262       This is a generic URI type with a fixed attribute to indicate
1263       that it should be resolved locally. URI is used instead of xs:id
1264       because name production rules in xs:id conflict with
1265       identifiers that may also be global handles or URNs.
1266     </xs:documentation>
1267   </xs:annotation>
1268   <xs:simpleContent>
1269     <xs:extension base="xs:anyURI">
1270       <xs:attribute name="local" type="xs:boolean" fixed="true"/>
1271     </xs:extension>
1272   </xs:simpleContent>
1273 </xs:complexType>
1274 <xs:complexType name="langString" abstract="true">
1275   <xs:annotation>
1276     <xs:documentation xml:lang="en">
1277   </xs:documentation>
1278 </xs:annotation>
1279 <xs:simpleContent>
1280   <xs:extension base="xs:string">
1281     <xs:attribute name="lang" type="xs:language"/>
1282   </xs:extension>
1283 </xs:simpleContent>
1284 </xs:complexType>
1285 <!-- -->
1286 <!-- Attribute type definitions -->
1287 <!-- -->
1288 <xs:simpleType name="rollupMethodType">
1289   <xs:restriction base="xs:token">
1290     <xs:enumeration value="all"/>
1291     <xs:enumeration value="any"/>
1292     <xs:enumeration value="fraction"/>
1293     <xs:enumeration value="units"/>
1294     <xs:enumeration value="mean"/>
1295     <xs:enumeration value="other"/>
1296   </xs:restriction>
1297 </xs:simpleType>
1298 <!-- -->
1299 <!-- Extensibility support -->
1300 <!-- -->
1301 <xs:group name="grp.any">
1302   <xs:annotation>
1303     <xs:documentation>
1304       Any namespaced element from any namespace may be included
1305       within an "any" element. The namespace for the imported element
1306       must be defined in the instance, and the schema must be imported.
1307     </xs:documentation>
1308   </xs:annotation>
1309   <xs:sequence>
1310     <xs:any namespace="##other" processContents="strict" minOccurs="0"
1311     maxOccurs="unbounded"/>
1312   </xs:sequence>

```

1313 </xs:group>  
1314 </xs:schema>