

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 © 2006 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 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.ieeeeltsc.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 **Contents**

55	1 Overview	1
56	1.1 Scope	2
57	1.2 Purpose	3
58	2 References	4
59	3 Definitions	4
60	3.1 Abbreviations and acronyms	5
61	4 Conformance	6
62	4.1 Data instances	6
63	4.2 Smallest permitted maximum values	6
64	5 Conceptual model (informative).....	7
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	5.1.4 Data capture	12
70	5.1.5 Example applications	14
71	5.2 Data model overview	16
72	6 Data model.....	17
73	6.1 Reusable Competency Mape	17
74	6.1.1 Identifier	18
75	6.1.2 Title	19
76	6.1.3 Description	19
77	6.1.4 SRCM schema label	20
78	6.1.5 SRCM schema version	20
79	6.1.6 Referential	20
80	6.1.7 Metadata	21
81	6.1.8 Graph	22
82	6.1.9 Extensions	23
83	6.2 Data types	23
84	6.2.1 Node type	23
85	6.2.2 Any type	35
86	6.2.3 LangString type	36
87	6.2.4 Local identifier type	38
88	6.2.5 Long identifier type	38
89	6.2.6 ProficiencyScoreType	39
90	Annex A (informative) Bibliography	40
91	Annex B (informative) Sample XML Binding.....	40

92 **Draft Standard for Learning Technology—** 93 **Simple Reusable Competency Map**

94 **1 Overview**

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

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

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

124 **1.1 Scope**

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

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

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

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

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

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

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

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

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

153 This Standard specifically does not cover:

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

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

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

185 **1.2 Purpose**

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

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

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

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

203 — A Standard data element by which
204 Simple Reusable Competency Maps can be identified as globally unique among compliant
205 systems and repositories.

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

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

216 — Correspondence with the IEEE Learning Objects Metadata Standard (IEEE
217 1484.12.1) developed by a parallel group.

218 **2 References**

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

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

223 (TBD: 1484.12.1 LOM)

224 **3 Definitions**

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

228 (TBD)

229 LangString bag: A structured datatype that represents one or more character strings. A Lang-
230 String may include multiple semantically equivalent character strings that represent transla-
231 tions of the same meaning into different languages. See also: datatype.

232 competency: For this Standard, a competency is defined as any form of knowledge, skill, atti-
233 tude, ability or learning outcome that can be described in a context of learning, education or
234 training.

235 Note—The word competency here is to be interpreted in the most broad sense to include edu-
236 cational objectives (those things that are sought) and competency or competencies (those
237 things that are achieved). The word “competency” is also used to include all classes of things
238 that someone, or potentially something, can be competent in, although some communities of
239 practice use the word with nuance, for example limiting its use to skill and excluding knowl-
240 edge or understanding. In learning contexts, a competency to be achieved is often named “ob-
241 jective”. Competency information represented in a competency map may also include facets
242 of competency, such as cognitive, psychomotor and affective, or behavioral indicators of com-
243 petency.

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

246 value space: The set of values for a given datatype (ISO/IEC 11404:1996).

247 NOTE:--In this Standard, a value space is typically enumerated outright, or defined by reference
248 to another Standard or specification.

249 **3.1 Abbreviations and acronyms**

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

251 ADL Advanced Distributed Learning

252 IMS IMS Global Learning Consortium

253 ISO International Standards Organization

254 JTC Joint Technical Committee

255 LTSC Learning Technology Standards Committee

256 RCD Reusable Competency Definition

257 RDCEO IMS Reusable Definition of Competency or Educational Objective

258 SCORM Shareable Courseware Object Reference Model

259 W3C World Wide Web Consortium

260 XML Extensible Mark-up Language

261 SPM smallest permitted maximum

262 URI: Uniform Resource Identifier

263 URN: Uniform Resource Name

264 **4 Conformance**

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

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

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

271 **4.1 Data instances**

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

273 **4.2 Smallest permitted maximum values**

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

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

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

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

290 NOTES:

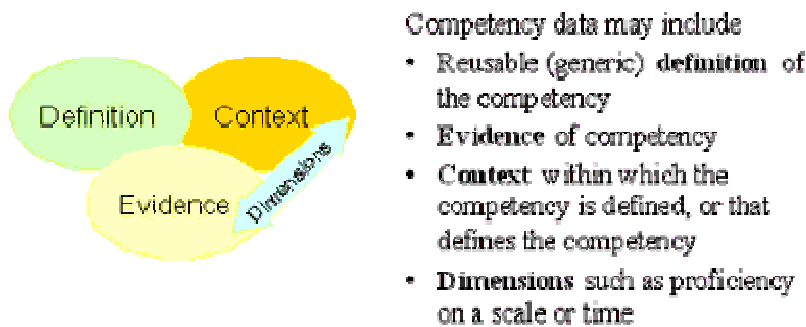
291 1—The intent is for the SPM values to cover most cases.

292 2—What "processing" means in the above depends on the nature of the application.

293 3—This Standard does not define any provision for how and whether a system can process more
294 than the SPM for a particular data element.

295 5 Conceptual model (informative)

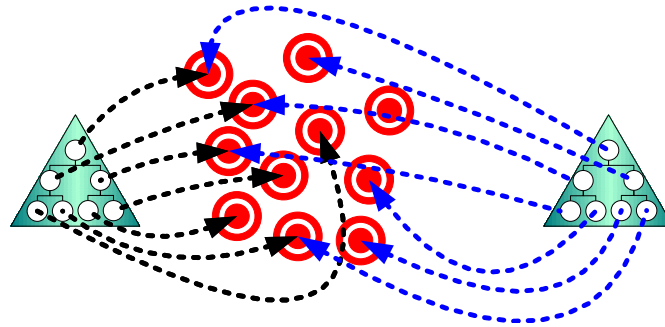
296 5.1 Functional overview



297

298 **Figure 1 —Reusable competency definitions capture only a part of the data that define a com-**
299 **petency**

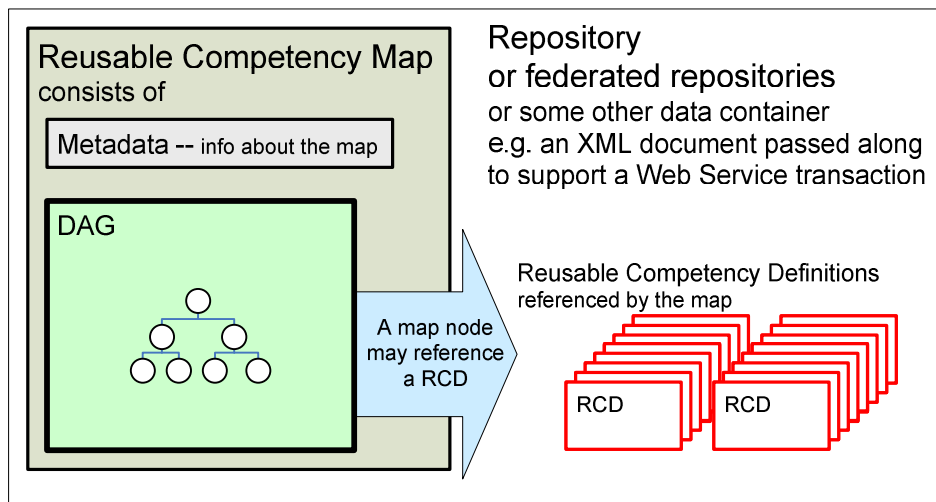
300 Reusable Competency Definitions (RCD), as defined in IEEE P1484.20.x, may used to cap-
301 ture a competency definition at any level of specificity, from the most precise to the most gen-
302 eral. The more specific a RCD is, the less reusable it is. Often a less precise definition is very
303 useful,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



311

312 **Figure 2 — Competency maps may represent different models of the same competency**313 **5.1.1 Modeling**

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

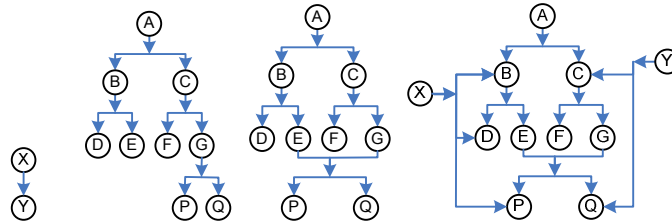
320 **5.1.2 Directed acyclic graph (DAG) map shape**

321

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

323 The Standard specifies the shape of a SRCM as a directed acyclic graph (DAG). A DAG is a
 324 hierarchical collection of nodes that implies containment. If node A has children B and C, it
 325 implies that A contains B or C. In competency terms, this would imply that competency A
 326 may be decomposed into sub-competencies or competency facets B and C, or that B and C
 327 contribute to A. In reality, many so-called existing competency “taxonomies” are not true tax-
 328 nomies, because the same competency components tend to appear more than once in the

329 model hierarchy. The Standard allows this to be represented in a DAG. The DAG allows a
 330 node to have more than one parent, as long as the parent cannot be a descendant of the node.
 331 Figure 4 shows four examples of DAGs, from simple to more complex. In the figure, arrows
 332 represent a parent-child relationship, with the arrows going from parent to child.



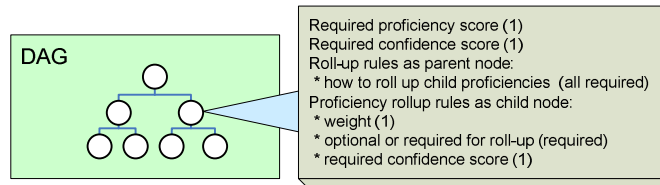
333

334

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

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

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



342

343

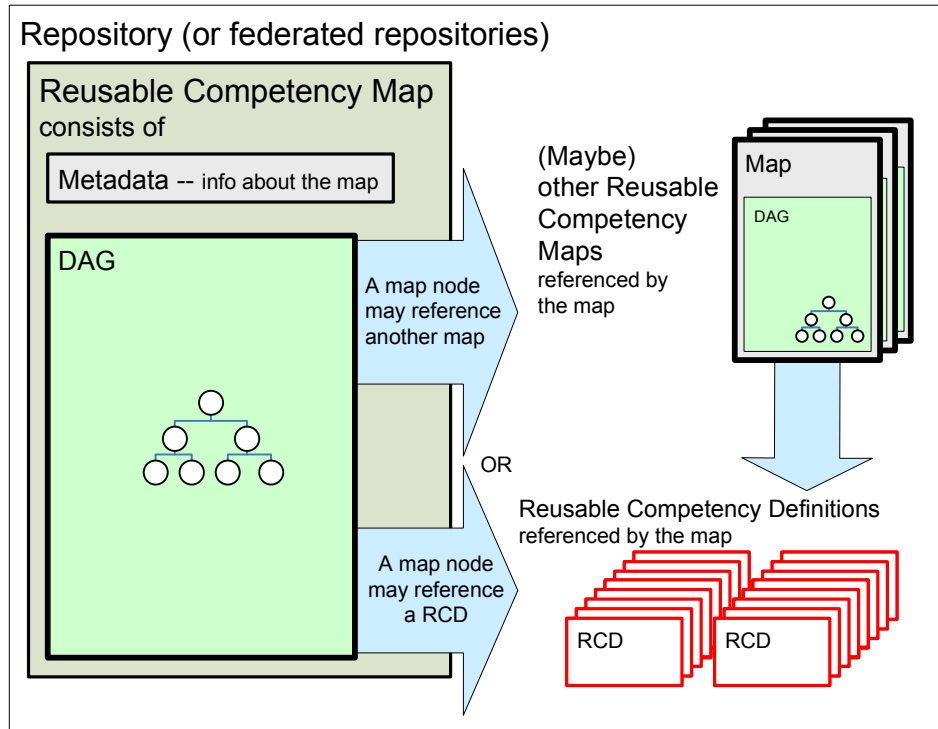
Figure 5 — Rules may be associated with nodes to define interpretations of the map

344 The fact that a rule is defined in the Standard does not imply that it is required in all applica-
 345 tions. For example, the required score required rules in Figure 5 may be required in an applica-
 346 tion profile for an assessment request, but may not be important when reporting a competency
 347 inventory for a person. In any case, the Standard defines a default value for each rule. For ex-
 348 ample, by default it is assumed that proficiency is required for all "children" of a defined com-
 349 petency in order to achieve proficiency for the "parent".

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

359 **5.1.3 Map extension by reference**

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

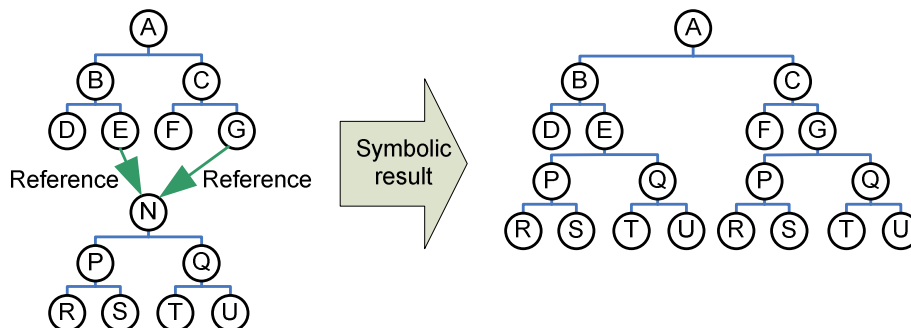


365

366

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

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



371

372

Figure 6 — Extension of a DAG map by reference to another DAG map

This is an unapproved proposal for a Standards Draft, subject to change.

373 5.1.3.1 Map extension rules and constraints

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

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

384 -- The referencing node retains its properties as child node, but the properties of the referenced
 385 node in its role as parent remain intact. In the example in Figure 6, nodes E and G must use the
 386 rollup rules defined by N for its children. If N references a RCD, and E also references a RCD,
 387 the RCDs are considered equivalent. In this case, when E is inspected in the context of its par-
 388 ent, i.e. when E is playing the role of child, the RCD that is visible is the RCD referenced by
 389 E. However, when E is inspected in the context of its children, the RCD that is visible is the
 390 RCD referenced by N, because E in its role as parent inherits the properties of N.

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

395 5.1.3.2 Using a DAG map to represent equivalencies and similarities

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



404

405

Figure 7 — A SRCM representing a one-way equivalency.

406 Figure 8 shows how a straightforward equivalency can be represented in a SRCM. Figure 7
 407 should be read as follows: Y is the only child of X, therefore proficiency in Y implies profi-
 408 ciency in X. Note that the reverse equivalency, if it is true, cannot be represented in the same
 409 SRCM because that would violate the acyclical constraint of a DAG. In practice, this is not a

410 problem, because competency equivalencies are often not reversible. For example, a country
 411 may accept diplomas from another country, but the reverse may not be true without some ad-
 412 ditional qualifications, as illustrated in Figure 8.

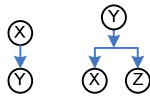
413 Figure 8 should be read as "Y has children X and Z. Therefore proficiency in Y requires profi-
 414 ciency in X as well as proficiency in Z".



415

416 **Figure 8— A SRCM representing a qualified reverse equivalency**

417 A common problem in dealing with competency modeling is imperfect equivalencies that are
 418 not reciprocal, as seen above for X and Y. The equivalency assumptions between competen-
 419 cies X and Y can however be expressed with two maps, as shown in Figure 9. Figure 9 should
 420 be read as follows: "Two SRCMs represent the relationships between competencies X and Y.
 421 Proficiency in Y implies proficiency in X. Proficiency in X implies proficiency in Y only if
 422 there is also proficiency in Z."



423

424 **Figure 9—Set of SRCMs representing the equivalency between competencies X and Y**

425 Note that the graphs in Figure 9 may not be merged into a single DAG, since that would vio-
 426 late DAG topology rules. It is the set of graphs that provides the semantics for the imperfect
 427 equivalency. In a practical application, it is likely that only one of the graphs would be used,
 428 since the available data would determine which node to use as a starting point to navigate the
 429 graph in a single direction. For example, in an operation to find which competency might en-
 430 compass X, the graph on the left would be rejected immediately since there is no parent for X,
 431 but the graph on the right can be navigated from X to find Y as an ancestor.

432 5.1.4 Data capture

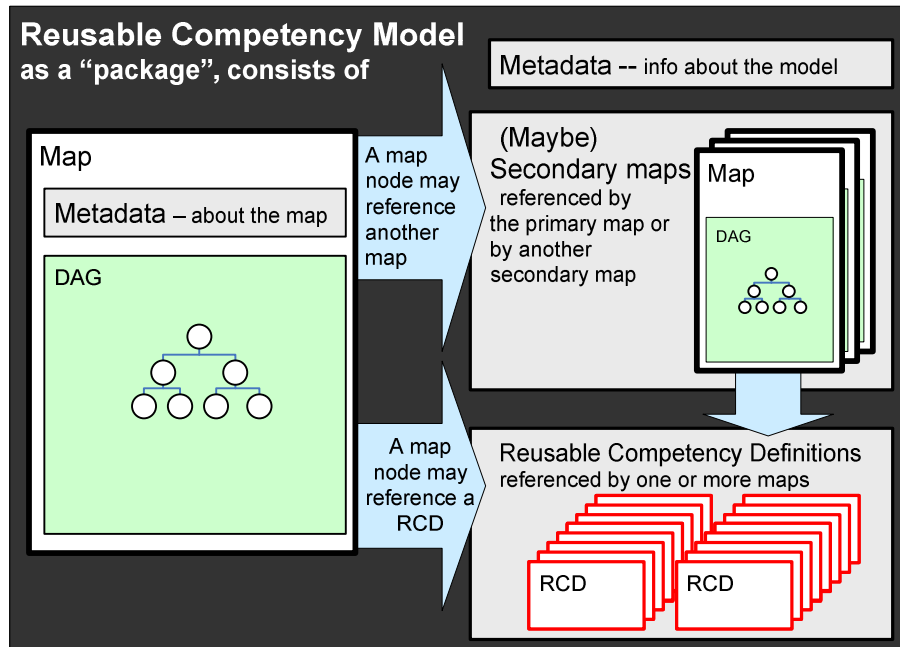
433 The data in a SRCM can come from many sources. Typical existing sources include various
 434 competency models developed by government agencies, academic institutions, enterprise and
 435 military training departments.

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

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

450 A cursory search of the World Wide Web turns up many hierarchical competency models
451 with considerable overlaps, many of which can be remapped easily into a SRCM instance
452 with associated RCDs. For example, an inventory of such models for the domain of military
453 leadership has been collected in the USA by the Air War College Center for Strategic Studies
454 -- <http://leadership.au.af.mil/sls-skil.htm#models> [Ed. Note: this to be fixed up as a proper ref-
455 erence]

456 Standards based technical documents such as S1000D conformant instances can also be mined
457 for the structure and content of a competency map. For example, a S1000D compliant techni-
458 cal manual for the maintenance and operation of a vehicle could be used as the source for
459 competency maps segmented for maintenance personnel and vehicle operators. In this exam-
460 ple, competency maps could also be updated automatically or semi-automatically as the de-
461 scribed equipment evolves. The application that extracts this information from the S1000D
462 instance could automatically guide the generation of new reusable competency definitions and
463 the creation or retirement of competency map nodes corresponding to feature additions or re-
464 movals in the vehicle's documentation.

465 **5.1.5 Example applications**

466

467

Figure 10— Application example: Packaging a job competency profile

468 A competency map can be used to represent the relationships between competency definitions
 469 in:

470 - a job competency profile (requirements for the job)

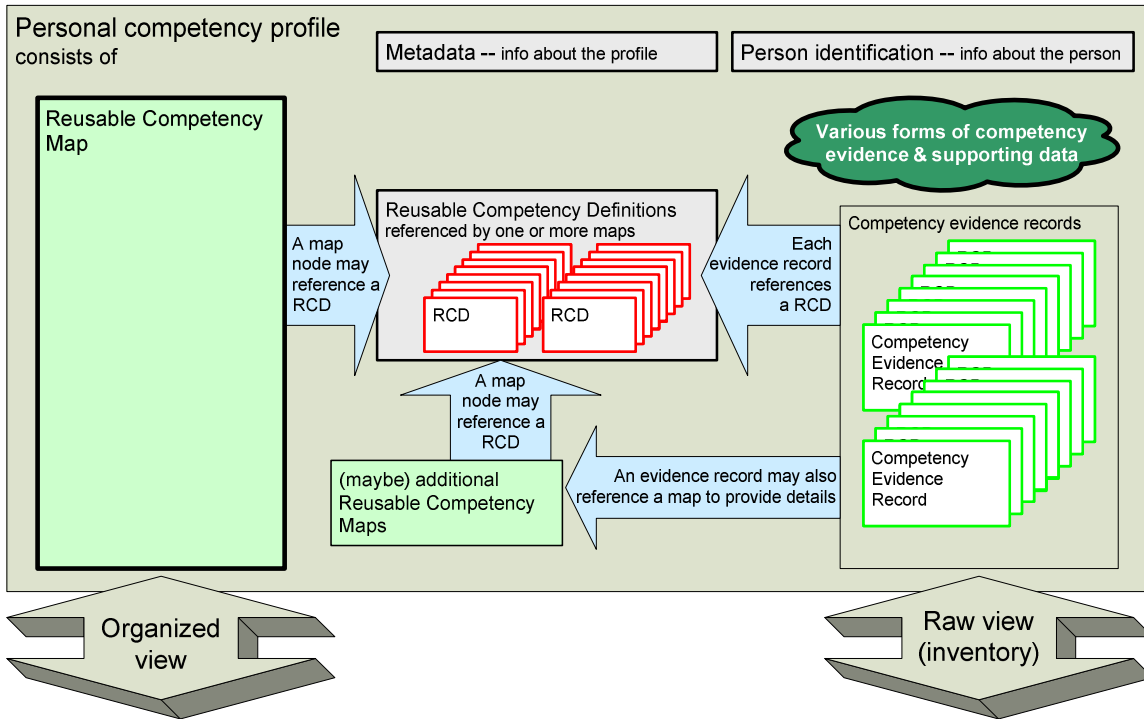
471 - a personal competency profile (acquired competencies)

472 - a personal competency gap profile (competencies to learn)

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

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

This is an unapproved proposal for a Standards Draft, subject to change.

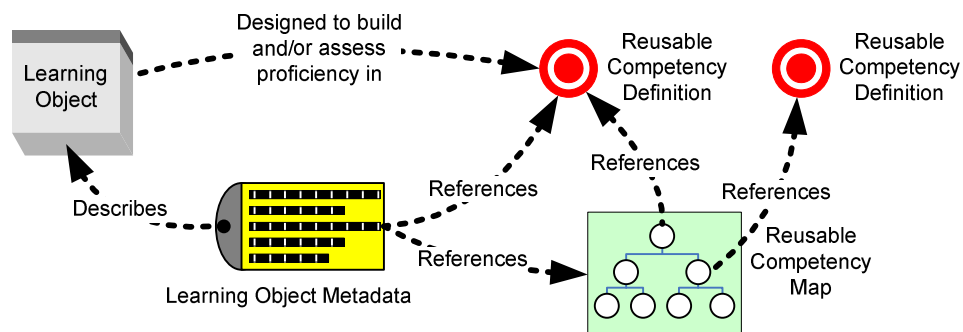


488

489

Figure 11 — Application example: A personal competency profile

490 Another example application would be in connection with learning object metadata, as shown
 491 in Figure 13. The competency taught by a learning object, or tested by an assessment object,
 492 should be represented by a reference to a reusable competency definition in the metadata. In
 493 IEEE LOM, this would be defined in one or more Classification elements. Someone looking
 494 for learning resources for a particular competency could then locate any competency map that
 495 contains a reference to the corresponding reusable competency definition, and then use that
 496 map to find higher level, related competency definitions. Searching for learning objects that
 497 specify those reusable competency definitions may then find learning objects that contain
 498 relevant resources, even when a one to one match on the original competency definition might
 499 fail. This can be made more reliable if the competency model assumed by the designers of the
 500 learning resource is made available in the metadata by referencing a reusable competency
 501 map.



502

503

Figure 12 — Application example: Finding related learning resources

504 **5.2 Data model overview**

505 The data model contains the following mandatory elements:

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

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

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

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

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

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

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

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

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

544 (End of informative clauses)

545 **6 Data model**

546 This Clause defines the data elements of a SRCM

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

548 NOTES:

549 1—The use of ISO/IEC 11404 notation in the synopses in 6.1 and 6.2 is for descriptive purposes
 550 only. A complete implementation of the operations defined in ISO/IEC 11404 is not required for
 551 conformance.

552 2—The ISO/IEC 11404 notation describes the semantics of the language-independent data types
 553 across all bindings (e.g., implementation of a data type as itself, its subtypes, its subclasses, and
 554 its specializations). For example, an ISO/IEC 11404 "record" may be implemented as an SQL
 555 table row, or as an XML complexType; an ISO/IEC 11404 "characterstring" may be implemented
 556 in an encoding (ISO 646, ASCII, ISO 8859-1, UTF-8, UTF-16, UTF-32, etc.) that supports the
 557 repertoire specified in the parameter to characterstring data type.

558 3—All examples in 6.1 and 6.2 are informative and do not endorse any particular binding.

559 4—The following language-independent data types used in this Standard are defined in ISO/IEC
 560 11404: bag, characterstring, record, set, state.

561 5—The labels for data elements and data types in the synopses in clauses 6.1 and 6.2 are for ref-
 562 erence only. There is no requirement that an implementation use the exact same labels, as long as
 563 the data elements and data types are semantically equivalent.

564 6—This Standard does not define a specific extension mechanism for the data model. Implemen-
 565 ters may define binding that allow additional elements, or create additional data models for com-
 566 petency data. Such models may be used to augment this model to support different communities
 567 of practice.

568 **6.1 Reusable Competency Mape**

569 **Synopsis**

```
570 reusableCompetencyMap :
571 record
572 (
573     identifier : longIdentifierType,
```

```

574     rcdRef : longIdentifierType,
575           // optional
576     title : langStringType(1000),
577           // the parameter value is the SPM
578           // optional
579     description : langStringType(4000),
580           // the parameter value is the SPM
581           // optional
582     srcmSchemaLabel: characterstring(iso-10646-1),
583           // SPM: 1000 characters
584     srcmSchemaVersion: characterstring(iso-10646-1),
585           // SPM: 1000 characters
586     referential: boolean,      metadata :      bag of anyType,
587           // SPM: 10 of any type in the bag
588     graph : ( see clause 6.1.8),
589     extensions : anyType
590 )

```

591 **Description**

592 The components of reusableCompetencyMap are defined in 6.1.1 – 6.1.9. Depending on
 593 the implementation, an instance of reusableCompetencyMap shall include one or more of
 594 the defined components. The map element is a required component. It may not be empty.

595 All top level elements in the reusableCompetencyMap data model are intrinsically unor-
 596 dered.

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

600 **6.1.1 Identifier**

601 **Synopsis**

```

602     identifier :
603         longIdentifierType,

```

604 **Description**

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

607 Subclause 6.2.1 defines longIdentifierType.

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

610 **6.1.2 Title**

611 **Synopsis**

```
612     title : bag of langStringType(1000),  
613         // SPM: 20 instance of langStringType in the bag  
614         // The parameter value is the SPM for the langString
```

615 **Description**

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

618 Subclause 6.3.1 defines langStringType.

619 NOTES:

620 1—The Title may be repeated in multiple languages.

621 2— While the Identifier provides the definitive reference to the SRCM, it is typically unintelligi-
622 ble. The Title provides a convenient alternative readable form, but one which is not the definitive
623 label. Examples: "English proficiency", "Schmiblick failure diagnostic level 4", "Demonstrates
624 conflict resolution skills".

625 **6.1.3 Description**

626 **Synopsis**

```
627     description : bag of langStringType(4000),  
628         // SPM: 20 instance of langStringType in the bag  
629         // The parameter value is the SPM for the langString
```

630 **Description**

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

634 Subclause 6.2.1 defines langStringType.

635 NOTES:

636 1—The Description may be repeated in multiple languages.

637 2—The description is typically more explicative than the title. Examples: "Proficiency in written
638 and spoken English and use of English for meaningful oral or written expression.", "Performance
639 of level 4 diagnostic as specified in IETM #SCMBLK007"

640 **6.1.4 SRCM schema label**

641 Synopsis

```
642     srcmSchemaLabel : characterstring(ISO-10646-1),  
643         // SPM: 1000 characters
```

644 **Description**

645 This element contains a label for the schema that defines and controls this SRCM
646 data instance. If this element is omitted then a value of “IEEE 1484.20???” (tbd) shall be as-
647 sumed. A conforming implementation may refuse to accept or process a SRCM instance if the
648 value of this data element is specified, but is not the value specified in the Standard.

649 NOTES

650 1—Different label values may be used to signal application profiles with particular extensions or
651 processing requirements.

652 2—This element does not describe the schema of the embedded metadata defined in 6.1.7. Every
653 instance of embedded metadata, if any, should include its own schema description.

654 **6.1.5 SRCM schema version**

655 Synopsis

```
656     srcmSchemaVersion : characterstring(ISO-10646-1),  
657         // SPM: 1000 characters
```

658 **Description**

659 Describes the version of the schema identified by `srcmSchemaLabel`. If this element is
660 omitted then a value of “1.0” shall be assumed. A conforming implementation may refuse to
661 accept or process a SRCM instance if the value of this data element is specified, but is not the
662 value specified in the Standard.

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

665 **6.1.6 Referential**

666 Synopsis

```
667     referential : boolean, // default = false
```

668 **Description**

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

673 Notes

674 1— The entity that creates the map is responsible for setting the value of this element to accu-
675 rately represent whether the map includes references to other maps.

676 2— This element is included to facilitate implementation efficiency in the processing of maps,
677 e.g. to help determine whether a map has external dependencies without having to examine
678 every node of the map.

679 **6.1.7 Metadata**

680 **Synopsis**

```
681     metadata :  
682         bag of anyType, // SPM: 10 of any type in the bag
```

683 **Description**

684 Optional embedded Metadata describing this SRCM.

685 If a metadata record is included, it is recommended that this record conform to IEEE 1484-
686 12.1-2002: Standard for Learning Object Metadata (IEEE LOM). In such conforming records,
687 the version of the Metadata Specification Standard is given in the meta-metadata element of
688 the metadata record. Some of the data elements defined in IEEE LOM are not relevant for a
689 SRCM and should be omitted. Profiles shall determine which metadata, if any, must be used
690 for conformance with the profile.

691 NOTES:

692 1—Useful metadata defined in the IEEE LOM include additional identification as an entry in one
693 or more catalogues, information about the author, publisher, the creation date, and the coverage
694 (in the sense of the Dublin Core as adopted by the IEEE LOM.) The Relation element may be
695 used to relate a definition to a prior version of the definition, and one or more Classification ele-
696 ments may be used to indicate where this particular map fits in a larger SRCM, map, model or
697 ontology of competencies or educational objectives. Classification elements may be repeated to
698 reference different models. For example, the SRCM might represent a map of a competency as
699 defined in O*NET model, as well as a map of a similar competency defined in a company's cus-
700 tom competency model.

701 2—More than one metadata record is allowed in the bag, but if there is more than one record each
702 record should conform to a different metadata specification. An implementation must accept any
703 metadata record that it cannot interpret, but it is not required to interpret such metadata records.

704 3—A particular binding specification or application profile may impose additional restrictions or
705 requirements.

706 6.1.8 Graph

707 Synopsis

```
708     graph : record :
709     (
710         defaultEntryNode: localIdentifierType,
711         entryNodes: bag of localIdentifierType,
712         nodes : bag of nodeType,
713         // SPM: (To be discussed. Specify in profiles?)
714     ),
```

715 Description

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

718 NOTES:

719 1— A binding might require a particular ordering of the elements. Such ordering does not imply
720 any meaningful order.

721 2— Some competency models prescribe an ordering or precedence for competencies. This typi-
722 cally means that competencies must be mastered in a specific order. This, in turn, means that
723 there are dependencies between the competencies so that mastery in A is a requirement for mas-
724 tery in B, B is a requirement for mastery of C, and so on. This can be described in the DAG by
725 making A a child of B, and B a child of C, etc. without adding an ordering property.

726 6.1.8.1 Default entry node

727 Synopsis

```
728     defaultEntryNode: localIdentifierType,
```

729 Description

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

733 6.1.8.2 Entry nodes

734 Synopsis

```
735     entryNodes: bag of localIdentifierType,
```

736 Description

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

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

741 **6.1.8.3 Nodes**

742 **Synopsis**

743 `nodes : bag of nodeType,`

744 **Description**

745 Unordered collection that contains all the nodes that make up the DAG. There is no intrinsic
746 order to the nodes, other than the order defined by parent-child relationships. NodeType is de-
747 fined in clause 6.2.1

748 **6.1.9 Extensions**

749 **Synopsis**

750 `extensions : anyType,`

751 **Description**

752 Application specific extensions to the data model. The extensions may not conflict with a
753 binding of the data model. Extensions may not replace or duplicate elements defined in the
754 data model.

755 NOTES:

756 1—The extensions should be defined by an application profile.

757 **6.2 Data types**

758 The following data types are used in conjunction with the data elements described in Clauses
759 6.1 and 6.2.

760 **6.2.1 Node type**

761 **Synopsis**

```
762     nodeType = record :
763     (
764         nodeId :
765             localIdentifierType,
766             // SPM: 1000 characters
767         rcdReference :
768             longIdentifierType,
769             // SPM: 4000 characters;
770         classLabel : record
```

This is an unapproved proposal for a Standards Draft, subject to change.

```

771         (
772             model : longIdentifierType,
773             token : characterstring,
774             bag of langStringType(250),
775             // SPM: 20 instances of langStringType in the bag
776             // The parameter value is the SPM for the
777             // langString
778         )
779     title :
780         bag of langStringType(1000),
781         // SPM: 20 instances of langStringType in the bag
782         // The parameter value is the SPM for the
783         // langString
784     description :
785         bag of langStringType(2000),
786         // SPM: 20 instances of langStringType in the bag
787         // The parameter value is the SPM for the
788         // langString
789     parents :
790         bag of localIdentifierType,
791         // nil if no parent
792     children :
793         bag of record
794         (
795             nodeRef: localIdentifierType,
796             weight: real(10,7), // range (0..1) default = 1,
797             dataRequired: boolean, // default = true;
798             proficiencyRequired: proficiencyScoreType,
799             proficiencyDesired: proficiencyScoreType,
800             // range (0..1) -- default = child's own value
801         )
802     symLink : longIdentifierType,
803             // must be nil or omitted if children is not nil
804     rules : record
805         (
806             proficiencyRequired: proficiencyScoreType,
807             proficiencyDesired: proficiencyScoreType,
808             rollupMethod : state
809                 (all,any,fraction,units,mean,other),
810             rollupParameter : choice(rollupMethod) of
811                 (
812                     all, any, mean : nil,
813                     fraction: real(10,7), // range (0..1),
814                     units: integer, // must be > 0
815                     other: characterstring(iso-10646-1),
816                     // SPM: 4000
817                 )
818         )
819 ),

```

820 **Description**

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

823 **6.2.1.1 Node Identifier**824 **Synopsis**

```
825     nodeId : localIdentifier,
```

826 **Description**

827 The value of this mandatory element is a unique label that identifies the node. The Identifier is
 828 sufficient to reference the node in the graph. This Identifier shall be unique at least in the con-
 829 text of the SRCM. There are no semantics associated with the value `nodeId` element; it is just a
 830 key to reference the node.

831 Subclause 6.x.xxx defines `localIdentifierType`.

832 **NOTES:**

833 1—If there is any possibility that the map may be disaggregated or merged with other maps or
 834 fragments of maps, it is recommended that the Identifier be globally unique. The result of merg-
 835 ing maps that contain nodes with conflicting identifiers is undefined.

836 **6.2.1.2 RCD Reference**837 **Synopsis**

```
838     rcdReference :  
839         longIdentifierType,  
840         // SPM: 4000 characters
```

841 **Description**

842 This data element contains the identifier of the Reusable Competency Definition represented
 843 by this node. If the node exists only for the purpose of grouping other nodes in the map, this
 844 element may be omitted or its value may be nil. There are no semantics associated with the
 845 value of `rcdReference` element. It is just a key to reference some Reusable Competency
 846 Definition, wherever it may be.

847 **6.2.1.3 Class Label**848 **Synopsis**

```
849     classLabel : record  
850         (  
851             model : longIdentifierType,  
852             bag of langStringType(250),
```

```

853         // SPM: 20 instances of langStringType in the bag
854         // The parameter value is the SPM for the
855         // langString
856     )

```

857 **Description**

858 This optional data element contains an optional reference to a model or vocabulary and a short
859 text label for the class of node. This element is provided to allow the provision of a class label
860 or similar label if the node is intended to represent a class in a particular competency model.

861 **NOTES**

862 1— An application may define that the model element is the identifier or locator for a vocabulary
863 specification that complies with the IMS VDEX specification.

864 2— This element allows the representation of the mapping to various hierarchical models, such as
865 Terminal learning objective (TLO) enabled by Enabling Learning Objectives (ELO)->ELO, Job
866 competencies implying task competencies, behaviors vs. knowledge, and so on. The specification
867 of node classes is outside the scope of this standard.

868 3— This element is multilingual to support the meaningful display of the node class in user inter-
869 faces. An application may define that only one language is meaningful for the label, in which case
870 the label may be treated as a language-independent token.

871 **6.2.1.4 Title**

872 **Synopsis**

```

873     title : bag of langStringType(1000),
874         // SPM: 20 instance of langStringType in the bag
875         // The parameter value is the SPM for the langString

```

876 **Description**

877 This optional data element contains a short human readable text label for the node.

878 If the node references a Reusable Competency Definition and this element has a value, im-
879 plementations shall display the title as defined by this element instead of the title of the refer-
880 enced Reusable Competency Definition. If the node references a Reusable Competency Defi-
881 nition and this element has no value, implementations should display the title as defined by the
882 referenced Reusable Competency Definition. However, this is not a requirement because per-
883 formance or other considerations may make this impractical. If no title is available for display,
884 but it is necessary to display the node, the implementation shall define what to display.

885 **6.2.1.5 Description**

886 **Synopsis**

```

887     description : bag of langStringType(2000),
888         // SPM: 20 instance of langStringType in the bag

```

889 // The parameter value is the SPM for the langString

890 **Description**

891 This optional data element contains a human readable text description for the node.

892 If the node references a Reusable Competency Definition and this element has a value, im-
893 plementations shall display the description as defined by this element instead of the descrip-
894 tion in the referenced Reusable Competency Definition. If the node references a Reusable
895 Competency Definition and this element has no value, implementations should display the
896 description as defined by the referenced Reusable Competency Definition, if available. How-
897 ever, this is not a requirement because performance or other considerations may make this im-
898 practical.

899 **6.2.1.6 Parents**

900 **Synopsis**

901 parents : bag of longIdentifierType,

902 **Description**

903 Identifiers of the parent nodes of this node in the directed graph topology. This element shall
904 be null, empty or omitted if the node does not have any parent. If a node has no parent, it shall
905 be included in the list of entry nodes (See Clause 6.1.8.2).

906 **NOTES**

907 1— In a DAG, a node may have more than one parent. A corollary is that several nodes may
908 share one or more children.

909 2— A binding may require the addition of a placeholder element to contain each parent reference
910 in the bag.

911 3— Although this is not recommended, a node that has one or more parent may be included in the
912 list of entry nodes.

913 **6.2.1.7 Children**

914 **Synopsis**

```
915       bag of record
916       (
917           nodeRef: localIdentifierType,
918           weight: real(10,7), // range (0..1) default = 1,
919           dataRequired: boolean, // default = true;
920           proficiencyRequired: proficiencyScoreType,
921           proficiencyDesired: proficiencyScoreType,
922                // range (0..1) -- default = child's own value
923       )
```

924 **Description**

925 This data element represents the child nodes of this node in the directed graph topology. This
 926 element shall be null, empty or omitted if the node has no children. A node shall be prohibited
 927 to have children if a `symLink` to a different map is specified for the node.

928 **6.2.1.8 Child Node Reference**929 **Synopsis**

```
930     children :
931         bag of record
932     (
933         nodeRef: localIdentifierType,
934         ...
935     )
```

936 **Description**

937 This mandatory data element shall be a reference to another node in the same map instance.
 938 The reference may not be to an ancestor node of the node that owns the bag of children.

939 Note – If the bag of children is not empty, every child record in the bag must include a valid
 940 `nodeRef`. An ancestor node may not be referenced because that would result in an illegal loop
 941 in the DAG topology.

942 **6.2.1.9 Child - Weight**943 **Synopsis**

```
944     children :
945         bag of record
946     (
947         nodeRef: localIdentifierType,
948         weight: real(10,7), // range (0..1) default = 1,
949         ...
950     )
```

951 **Description**

952 This data element is used only when the node is a child in a rollup operation where a weight
 953 assigned to the information associated with the child node is relevant. For example, if the
 954 rollup method is mean, this data element can be used to specify a relative weight among chil-
 955 dren. In the absence of a specific value, the default value of this element is assumed to be 1.

956 Because the value of the `weight` element may be different in the context of different parents,
 957 this element is defined in a rule associated with the parent.

958 Note – This element allows certain rollup operations where proficiency information is summa-
 959 rized in such a way that weights may influence the result. Weights are not relevant in some of
 960 the rollup methods defined in Clause 6.2.1.13.

961 **6.2.1.10 Child – Data Required**

962 **Synopsis**

```
963     children :
964         bag of record
965     (
966         nodeRef: localIdentifierType,
967         ...
968         dataRequired: boolean, // default = true;
969         ...
970     )
```

971 **Description**

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

976 Because the value of the `required` element may be different in the context of different par-
 977 ents, this element is defined in a rule associated with the parent.

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

982 **6.2.1.11 Child - Proficiency Required**

983 **Synopsis**

```
984     children :
985         bag of record
986     (
987         nodeRef: localIdentifierType,
988         ...
989         proficiencyRequired: proficiencyScoreType,
990         ...
991     )
```

992 **Description**

993 This data item specifies a proficiency requirement for a child in the context of this parent
 994 node. It specifies that proficiency status can be assumed to be true if the available proficiency
 995 measure associated with the child node is at least a certain value. For valid comparisons, the

996 proficiency measure must be expressed in a compatible range. The range for profi-
 997 ciencyRequired is -1 to 1 inclusive. The default default value if not explicitly specified
 998 is 1.

999 The `proficiencyRequired` element defined in the child node's own rules shall be used if
 1000 no overriding value is specified in this data element.

1001 NOTES:

1002 1— Because the value of the `proficiencyRequired` element may be different in the context
 1003 of different parents, this data element allows `proficiencyRequired` to be specified in this
 1004 data element associated with the parent.

1005 2— Proficiency measure is also called "proficiency level", "score" or "success measure" in vari-
 1006 ous specifications and Standards.

1007 3— Application example involving rollup: Node RX references RCD X and specifies a profi-
 1008 ciency required of 70%. However, node RX is a child of node RA that specifies that for child RX
 1009 the proficiency required is 80%. When rolling up competency status information from RX into
 1010 the competency status information for RA, the proficiency required used to evaluate whether a
 1011 measure satisfies the requirement will be 80%. This is because the requirements specified for the
 1012 context override the requirements specified for individual components of the context. On the
 1013 other hand, if no rule associated with RA specifies a particular proficiency required for child RX,
 1014 then the proficiency required defined in the node RX is used by default.

1015 **6.2.1.12 Child - Proficiency Desired**

1016 **Synopsis**

```
1017     children :
1018         bag of record
1019         (
1020             nodeRef: localIdentifierType,
1021             ...
1022             proficiencyDesired: proficiencyScoreType
1023         )
```

1024 **Description**

1025 This data item specifies a proficiency requirement for a child in the context of this parent
 1026 node. It specifies that proficiency status can be assumed to be true if the available proficiency
 1027 measure associated with the child node is at least a certain value. For valid comparisons, the
 1028 proficiency measure must be expressed in a compatible range. The range for `proficiency-`
 1029 `Desired` is -1 to 1 inclusive. The default default value if not explicitly specified is 1.

1030 The `proficiencyDesired` element defined in the child node's own rules shall be used if
 1031 no overriding value is specified in this data element.

1032 NOTE— Application example: The current node references RCD X and a competency evidence
 1033 record for an individual also points to RCD X, and specifies that the individual's proficiency level

1034 is 67%. If a `proficiencyRequired` of 60% (=0.6) is specified by for this node, the individual
 1035 is not considered proficient according to this competency map. However, if a proficiency `profi-`
 1036 `ciencyDesired` of 75% is specified, then a learning application might try to engage the indi-
 1037 vidual in a learning activity to achieve the desired level of proficiency, even though the baseline
 1038 required proficiency is already achieved.

1039 **6.2.1.13 Sym Link**

1040 **Synopsis**

1041 `symLink` : `longIdentifierType`,

1042 **Description**

1043 Identifier referencing a map used to extend the current map by merging an entry node in that
 1044 map with the current node. The binding of the identifier may include the specification of a par-
 1045 ticular entry node in the target map. If the target map has multiple entry nodes and no entry
 1046 node is specified in this element, the default entry node in the target map shall be used as entry
 1047 node. The merging rules are specified in clause x.x (TBD).

1048 **NOTES:**

1049 1— See conceptual overview for a detailed informative explanation of the map extension mecha-
 1050 nism.

1051 2— By using sym links, it is possible to keep SRCMs to an easily manageable size, and to reuse
 1052 SRCMs in the context of other maps. For example, the same skill definition published by a state
 1053 standards board may be included in various institution-specific competency models through this
 1054 mechanism.

1055 **SymLink merging rules**

1056 When a sym link is used, the target node that is the entry point in the target map shall be
 1057 treated as if it was merged with the referencing node when processing data from the tree. If
 1058 any two data elements or attributes conflict between the nodes during processing, the element
 1059 or attribute value of the referencing node shall override the value of the target node. Children
 1060 of the target node, if any, shall be treated as children of the referencing node when traversing
 1061 the graph and performing rollups. Deleting a referencing node shall have no effect on the tar-
 1062 get node. Changing values or data elements of the referencing node shall have no effect on the
 1063 values or data elements of the target node. Changing values or data elements of the target node
 1064 shall have no effect on the values or data elements of the referencing node. The only actual
 1065 merging or override of values or data elements shall occur in the data space of the processing
 1066 entity.

1067 **6.2.1.14 Rules**

1068 **Synopsis**

1069 `rules` : `record`

```

1070     (
1071         proficiencyRequired: proficiencyScoreType,
1072         proficiencyDesired: proficiencyScoreType,
1073         rollupMethod : state(all,any,fraction,units,mean,other),
1074         rollupParameter : choice(rollupMethod) of
1075             (
1076                 all, any, mean : nil,
1077                 fraction: real(10,7), // range (0..1),
1078                 units: integer, // must be > 0
1079                 other: characterstring(iso-10646-1),
1080                 // SPM: 4000
1081             )
1082     )

```

1083 **Description**

1084 Rules that specify the data processing behavior of systems that use the SRCM.

1085 If the rules element is not present in an instance, the default value for each the rule elements
 1086 defined here shall be applied in processing rules. If the rules element is present but any of the
 1087 rule elements is not specified, the default value for that element shall be assumed in processing
 1088 rules.

1089 **6.2.1.15 Proficiency required**

1090 **Synopsis**

```
1091     proficiencyRequired: proficiencyScoreType,
```

1092 **Description**

1093 This data item specifies a proficiency requirement. It specifies that proficiency status can be
 1094 assumed to be true if the available proficiency measure for the referenced competency defini-
 1095 tion is at least a certain value. For valid comparisons, the proficiency measure must be ex-
 1096 pressed in a compatible range. The range for `proficiencyRequired` is -1 to 1 inclusive.
 1097 The default default value if not explicitly specified is 1.

1098 The `proficiencyRequired` element defined in the node's rules shall be used if the node
 1099 is considered independently of any parent or if no overriding value is specified in the rules of
 1100 the parent node being considered in an operation. Because the value of the `profi-`
 1101 `ciencyRequired` element may be different in the context of different parents, a value for
 1102 `proficiencyRequired` may also be specified in the child specification rule associated
 1103 with each parent of this node. For rollup processes, if a value for `proficiencyRequired` is
 1104 specified in the child specification of the parent node, that value shall be used instead of the
 1105 value of the `proficiencyRequired` element of the child node.

1106 NOTES:

1107 1— Proficiency measure is also called "proficiency level", "score" or "success measure" in vari-
1108 ous specifications and Standards. Some semantic mapping may be required for integration with
1109 those specifications and standard.

1110 2— Application example: The current node references RCD X and a competency evidence record
1111 for an individual also points to RCD X, and specifies that the individual's proficiency level is
1112 67%. If a proficiency required of 70% (=0.7) is specified by for this node, the individual is not
1113 considered proficient according to this competency map.

1114 3— Application example involving rollup: Node RX references RCD X and specifies a profi-
1115 ciency required of 70%. However, node RX is a child of node RA that specifies that for child RX
1116 the proficiency required is 80%. When rolling up competency status information from RX into
1117 the competency status information for RA, the proficiency required used to evaluate whether a
1118 measure satisfies the requirement will be 80%. This is because the requirements specified for the
1119 context override the requirements specified for individual components of the context. On the
1120 other hand, if no rule associated with RA specifies a particular proficiency required for child RX,
1121 then the required defined for RX is used by default.

1122 **6.2.1.16 Proficiency desired**

1123 **Synopsis**

1124 `proficiencyDesired: proficiencyScoreType,`

1125 **Description**

1126 This data item specifies a proficiency requirement. It specifies that proficiency status can be
1127 assumed to be true if the available proficiency measure for the referenced competency defini-
1128 tion is at least a certain value. For valid comparisons, the proficiency measure must be ex-
1129 pressed in a compatible range. The range for `proficiencyDesired` is -1 to 1 inclusive.
1130 The default default value if not explicitly specified is 1.

1131 The `proficiencyDesired` element defined in the node's rules shall be used if the node is
1132 considered independently of any parent or if no overriding value is specified in the rules of the
1133 parent node being considered in an operation. Because the value of the `proficiencyDe-`
1134 `sired` element may be different in the context of different parents, a value for `profi-`
1135 `ciencyDesired` may also be specified in the child specification rule associated with each
1136 parent of this node. For rollup processes, if a value for `proficiencyDesired` is specified
1137 in the child specification of the parent node, that value shall be used instead of the value of the
1138 `proficiencyDesired` element of the child node.

1139 NOTE— Application example: The current node references RCD X and a competency evidence
 1140 record for an individual also points to RCD X, and specifies that the individual's proficiency level
 1141 is 67%. If a proficiency required of 60% (=0.6) is specified by for this node, the individual is not
 1142 considered proficient according to this competency map. However, if a proficiency desired of
 1143 75% is specified, then a learning application might try to engage the individual in a learning ac-
 1144 tivity to achieve the desired level of proficiency, even though the baseline required proficiency is
 1145 already achieved.

1146 **6.2.1.17 Rollup method**

1147 **Synopsis**

1148 `rollupMethod : state(all, any, fraction, units, mean, other),`

1149 **Description**

1150 This data item applies only when the node is a parent. It specifies how to roll up proficiency
 1151 information specified by children of the node. The defined methods are:

- 1152 • **all** -- Proficiency must be achieved for all the competencies represented by chil-
 1153 dren nodes in order to consider that proficiency is achieved for the competency repre-
 1154 sented by this node. This is the assumed default value if a rollup method is not speci-
 1155 fied.
- 1156 • **any** -- Proficiency must be achieved for any of the competencies represented by
 1157 children nodes in order to consider that proficiency is achieved for the competency
 1158 represented by this node.
- 1159 • **fraction** -- Proficiency must be achieved for at least specified fraction of the compe-
 1160 tencies represented by children nodes in order to consider that proficiency is achieved
 1161 for the competency represented by this node. The fraction value is expressed as a
 1162 floating point number in the range 0 to 1 inclusive. 0 means that no child proficiency is
 1163 required. 1 is the equivalent of specifying that the rollup method is all. 0.5 means that
 1164 proficiency must be achieved for at least 50% of the child nodes.
- 1165 • **units** -- Proficiency must be achieved for at least the specified number of the compe-
 1166 tencies represented by children nodes in order to consider that proficiency is achieved
 1167 for the competency represented by this node. The number is a positive integer. 0
 1168 means that no child proficiency is required. 1 is the equivalent of specifying that the
 1169 rollup method is any.
- 1170 • **mean** -- The proficiency measure for the parent node is determined by averaging the
 1171 proficiency levels for the competencies represented by children nodes. The proficiency
 1172 status for the parent node can then be determined by comparing that level with the pro-
 1173 ficiencyRequired value for the parent node. If no proficiency level is available for a
 1174 child node, the proficiency status (boolean) is used instead, with proficient = 1 and not
 1175 proficient or unknown = 0.

- 1176 • other -- Another proficiency method is to be used. The other method must be speci-
1177 fied by an application profile and is not defined by the Standard.

1178 By default, the rollup method is "all". No available competency data or a value of "unknown"
1179 is considered the same as "not proficient" for the purpose of rollup. If proficiency status but no
1180 proficiency measure is available for any node, the proficiency measure value is assumed to be
1181 1 if proficient, and 0 if not proficient.

1182 NOTES:

1183 1— Proficiency measure is also called "proficiency level", "score" or "success measure" in vari-
1184 ous specifications and Standards.

1185 **6.2.1.18 Rollup parameter**

1186 **Synopsis**

```
1187     rollupParameter : choice(rollupMethod) of
1188     (
1189         all, any, mean : nil,
1190         fraction: real(10,7), // range (0..1),
1191         units: integer, //
1192         other: characterstring(iso-10646-1),
1193         // SPM: 4000
1194     )
```

1195 **Description**

1196 Parameter used only if the specified rollup method is fraction, units or other. The type of pa-
1197 rameter depends on the rollup method.

1198 If the rollup method is "other", the parameter is a container for data of any type, and an appli-
1199 cation profile must specify the method and the data.

1200 **6.2.2 Any type**

1201 **Synopsis**

```
1202     type anyType = (unspecified);
```

1203 **Description**

1204 This data type represents any type not specified in this Standard. This Standard does not re-
1205 quire an implementation to process data elements of this type when encountered in a RCD
1206 instance. The data types implemented in a SRCM data element with type anyType shall not
1207 preclude interoperable bindings of the entire SRCM instance.

1208 NOTES

1209 1— If implementations specify or require data elements for which the type is defined in this Stan-
 1210 dard as anyType, this Standard recommends that the implementations provide the means to inter-
 1211 pret and validate the implementation-specific data. For example, an implementation that uses an
 1212 XML binding should include a valid XML schema that can be referenced in RCD instances
 1213 bound in XML documents, and the schema should be documented.

1214 2—The data types used in the container should be defined by an application profile.

1215 Examples

1216 An application profile might specify an additional data element for "desired proficiency" to use
 1217 along with "required proficiency" in SRCM instances that represent a position competency model
 1218 for hiring.

1219 An application profile might specify a rollup method that is not defined in the standard, and em-
 1220 bed the specification for that rollup method as an extension in a standard-conformant instance.

1221 An application profile might add references to contextual information that can be used to make an
 1222 instance of the standard data model more context-specific.

1223 An application profile might add metadata to identify the identity that set a proficiency level re-
 1224 quirement, and to reference the enterprise policy that is the source of the requirement.

1225 6.2.3 LangString type

1226 Synopsis

```
1227     type langStringType(length) =
1228         record
1229         (
1230             language :
1231                 languageType,
1232             string :
1233                 characterstring(iso-10646-1)
1234                 // SPM: The length parameter
1235         );
```

1236 Description

1237 This data type consists of a language specification for a string and the string itself.

1238 Examples

1239 The following are three examples of localized strings: "Information Technology" in
 1240 French, "localization" in British English, and "xxx" in Japanese hiragana.

```
1241     ( "fr", "Technologies de l'information" ),
1242     ( "en-GB", "localisation" ),
1243     ( "jp-JP-jisx208", "xxx" ),
```

1244 **6.2.3.1 Language**1245 **Synopsis**

```
1246     language :
1247         characterstring(iso-646),
1248         // SPM: 250 characters
```

1249 **Description**

1250 The `language` data element specifies the language of the localized string. The format of this
 1251 data type is a character string consisting of a required language code followed by multiple,
 1252 optional, hyphen-prefixed subcodes (see examples below).

1253 The following rules apply to the language code part of the character string :

- 1254 – 2-letter codes are defined by ISO 639–1.
- 1255 – 3-letter codes are defined by ISO 639–2.
- 1256 – The value prefix "i" is reserved for registrations defined by the Internet Assigned
 1257 Numbers Authority (IANA).
- 1258 – The value prefix "x" is reserved for private use.

1259 The following rules apply to the first subcode part of the character string :

- 1260 – 2-letter subcodes are ISO 3166–1 alpha-2 country codes.
- 1261 – Subcodes of from 3 to 8 letters are registered with IANA.

1262 Rules for additional subcodes are unspecified, except that the length of any subcode cannot
 1263 exceed 8 characters.

1264 NOTE—The language code is often given in lower case and the subcodes (if any) in upper case.
 1265 However, the values are case insensitive.

1266 **Examples**

```
1267     "en-GB"
1268     "de"
1269     "fr-CA"
1270     "it"
1271     "grc" (Ancient Greek, until 1453)
1272     "en-US-NY-NY"
1273     "eng-GB-cockney"
1274     "map-PG-buin" (Austronesian - Papua New Guinea Buin)
1275     "gem-US-PA"
```

1276 **6.2.3.2 String**1277 **Synopsis**

```
1278     string :
1279         characterstring(iso-10646-1),
1280         // SPM: The length parameter
```

1281 **Description**1282 The `string` data element contains the text of the localized string.1283 **6.2.4 Local identifier type**1284 **Synopsis**

```
1285     type localIdentifierType = characterstring(iso-10646-1)
1286         // SPM: 4000 characters
```

1287 **Description**

1288 This data type is an identifier (a label) associated with an object that is intended to be unique
 1289 within the context of usage of the object. The character string shall conform to the syntax for
 1290 Uniform Resource Identifiers (URIs) as defined by RFC 2396.

1291 **NOTES**

1292 1—This Standard recommends that the URI be a globally unique identifier in the form of a Uni-
 1293 form Resource Name (URN) (see RFC 2141 [A5]) or a Handle as defined in the Handle System
 1294 [Ax].

1295 2— Depending on the binding, these data elements may be implemented as explicit or implicit.
 1296 The binding may be a combination of explicit and implicit representations of these data elements.
 1297 For example, in an XML document they may be implicit in the structure of the document if the
 1298 shape of the graph is a simple tree.

1299 3— Depending on the implementation, the values in the parents and children collections may
 1300 be implicit, may be explicit and literal or may be an implementation-specific identifier or
 1301 pointer. However, any conforming implementation of a Reusable Competency Map must rep-
 1302 resent no less than the information specified in this Standard for any component that is present
 1303 in the implementation. For example, even it is a node is represented by a relative pointer to
 1304 achieve data compression in a particular binding, the identifier for that node must be resolv-
 1305 able to a conformant identifier type when the tree instance is transmitted to another confor-
 1306 mant implementation.

1307 **6.2.5 Long identifier type**1308 **Synopsis**

```
1309     type longIdentifierType =
1310         record (
1311             catalog: characterstring(iso-10646-1),
1312                 // SPM: 4000 characters
1313             entry: characterstring(iso-10646-1)
1314                 // SPM: 4000 characters
1315         );
```

1316 Description

1317 This data type is an identifier (a label) associated with an object that is intended to be unique
1318 within the context of usage of the object. The character string shall conform to the syntax for
1319 Uniform Resource Identifiers (URIs) as defined by RFC 2396.

1320 NOTE—This Standard recommends that the URI be a globally unique identifier in the form of a
1321 Uniform Resource Name (URN) (see RFC 2141 [A5]) or a Handle as defined in the Handle Sys-
1322 tem [Ax].

1323 6.2.6 ProficiencyScoreType**1324 Synopsis**

```
1325     type proficiencyScoreType = real(10,7);  
1326     // range -1..1
```

1327 Description

1328 This data type is a proficiency score on a continuous scale constrained to the values -1 to 1
1329 inclusive. Negative scores are allowed to permit the representation of "negative proficiency",
1330 e.g. if a person is actually dangerously inept in the execution of a task.

1331 NOTES

1332 1— Negative scores are allowed to permit the representation of "negative proficiency", e.g. if a
1333 person is actually dangerously inept in the execution of a task.

1334 2— This type maps exactly to the scaled score type defined in IEEE 1484.11.1.

1335 3— Rollups and comparisons of scores from different sources require a common scale. Various
1336 proficiency scales or grading methods may be mapped to and from the ProficiencyScoreType. For
1337 discontinuous scales, the mapping typically involves ranges or thresholds. Such mappings are
1338 outside the scope of this standards.

1339

1340 **Annex B**

1341 [A2] IEEE 1484.11.2–2003, Standard for Learning Technology—ECMAScript Application
1342 Programming Interface for Content to Runtime Services Communication.

1343 [A3] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

1344 [A5] RFC 2141, "URN Syntax," Network Working Group, May 1997.

1345 **IMSRDCEO** IMS Global Learning Consortium, Inc. (2002). IMS Reusable Defini-
1346 tion of Competency or Educational Objective, Version 1.0

1347 **O*NET** <http://online.onetcenter.org/> (or
1348 http://www.access.gpo.gov/o_net/datadict/datadict.pdf)

1349 **Ostyn** <http://ltsc.ieee.org/doc/wg20/CompDefInit.doc> (Base document from
1350 P1484.20)

1351 **PASS** <http://www.ous.edu/pass/Standards/admission.html> (Oregon Profi-
1352 ciency-based Admissions Standards System)

1353 **SCANS** <http://www.tier.net/tcenters/scans.htm> (Secretary's Commission on
1354 Achieving Necessary Skills: Competencies)

1355 **SCORM** <http://www.adlnet.org> (ADL SCORM)

1356 **TATS** <http://www.adtdl.army.mil/atdls.htm> (Total Army Training System)

1357 **Annex B**

1358 (informative)

1359 Sample XML Binding

1360 This sample shows an XML instance and the corresponding schema for an XML binding.
 1361 This is a sample only. The instance shows that RCDs are known to exist for 3 competencies
 1362 defined by different agencies: "Can drive a car safely", "Can steer a car and control its speed"
 1363 and "Knows and understands road signs and markings", and that the first competency is com-
 1364 posed of the other two. No rules are specified because the default rules are good enough.

```

1365 <?xml version="1.0" encoding="UTF-8"?>
1366 <simpleCompetencyMap xmlns="proposedForIEEE-LTSC-WG20/simpleReusableCompetencyMap"
1367 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1368 xsi:schemaLocation="proposedForIEEE-LTSC-WG20/simpleReusableCompetencyMap simple-
1369 competency-map-draft.xsd">
1370   <mapId>123356/123.45tyu/345xyz</mapId>
1371   <schemaLabel>SRCM0.1.0</schemaLabel>
1372   <schemaVersion>1.0</schemaVersion>
1373   <graph>
1374     <defaultEntryNode nodeRef="someIdOrOther6"/>
1375     <entryNodes>
1376       <entryNode nodeRef="someIdOrOther6"/>
1377     </entryNodes>
1378     <node nodeId="someIdOrOther6">
1379       <rcdRef ref="http://somelicensingdept.gov/ref/driver/minimal"/>
1380       <title>
1381         <langString lang="en">Can drive a car safely</langString>
1382         <langString lang="fr">Capable de conduire une voiture prudemment</langString>
1383       </title>
1384       <children>
1385         <child nodeRef="someIdOrOther7"/>
1386         <child nodeRef="someIdOrOther8"/>
1387       </children>
1388     </node>
1389     <node nodeId="someIdOrOther7">
1390       <rcdRef ref="http://somelicensingdept.gov/ref/driver/sk1"/>
1391       <title>
1392         <langString lang="en">Can steer a car and control its speed</langString>
1393         <langString lang="fr">Peut diriger et controler la vitesse</langString>
1394       </title>
1395       <parents>
1396         <parent nodeRef="someIdOrOther6"/>
1397       </parents>
1398     </node>
1399     <node nodeId="someIdOrOther8">
1400       <rcdRef ref="http://usdot.gov/rcdsroadsigns2005"/>
1401       <title>
1402         <langString lang="en">Knows and understands road signs and markings</langString>
1403         <langString lang="fr">Connait et comprend les signaux routiers</langString>
1404       </title>
1405       <parents>
1406         <parent nodeRef="someIdOrOther6"/>
1407       </parents>
1408     </node>
1409   </graph>
1410 </simpleCompetencyMap>

```

1411

```

1412 <?xml version="1.0" encoding="UTF-8"?>
1413 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:srcm="proposedForIEEE-LTSC-
1414 WG20/simpleReusableCompetencyMap" targetNamespace="proposedForIEEE-LTSC-
1415 WG20/simpleReusableCompetencyMap" elementFormDefault="qualified"
1416 attributeFormDefault="unqualified" version="experimental0.4">
1417   <xs:annotation>
1418     <xs:documentation>
1419       Copyright 2005, 2006 Claude Ostin - All rights reserved.
1420       Schema for a simple reusable competency map implementing a proposed standard.

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1421     See proposed standard text at http://ostyn.com/resources.htm for details.
1422     A competency map may reference a Reusable Competency Definition
1423     (RCD) or be standalone.
1424     The map represents a hierarchy of Competency Definitions. It does not include
1425     the RCDs. When used as a map of RCDs, different map instances
1426     may represent different beliefs as to how those RCDs are related.
1427     The nodes in the map are organized as a Directed Acyclic Graph (DAG),
1428     but the graph can be easily used to represent a tree with a single root node.
1429     The graph is represented as a flat list of nodes,
1430     each of which has a parent and/or and children list (which may be empty).
1431     The human-readable strings in the map may be specified in multiple
1432     languages. This introduces a very small amount of overhead if you
1433     do not need that feature, e.g. instead of something like
1434         (title)my title(/tilte)
1435     you use
1436         (title)
1437         (langString)my title(/langString)
1438         (/title)
1439     This is a strawman schema and should not be used for conformance.
1440     </xs:documentation>
1441 </xs:annotation>
1442 <!-- -->
1443 <!-- Reusable elements -->
1444 <!-- -->
1445 <xs:element name="simpleCompetencyMap" type="srcm:competencyDAGGraphType">
1446   <xs:annotation>
1447     <xs:documentation>
1448       Complete map, consisting of front matter and the actual graph.
1449     </xs:documentation>
1450   </xs:annotation>
1451 </xs:element>
1452 <xs:element name="classLabel" type="srcm:classLabelType">
1453   <xs:unique name="UniqueClassLabelLanguageInstance">
1454     <xs:selector xpath="./srcm:langString"/>
1455     <xs:field xpath="@lang"/>
1456   </xs:unique>
1457 </xs:element>
1458 <xs:element name="description" type="srcm:langStringBagType">
1459   <xs:unique name="UniqueDescriptionLanguageInstance">
1460     <xs:selector xpath="./srcm:langString"/>
1461     <xs:field xpath="@lang"/>
1462   </xs:unique>
1463 </xs:element>
1464 <xs:element name="rcdRef" type="srcm:rcdRefType"/>
1465 <xs:element name="title" type="srcm:langStringBagType">
1466   <xs:unique name="UniqueTitleLanguageInstance">
1467     <xs:selector xpath="./srcm:langString"/>
1468     <xs:field xpath="@lang"/>
1469   </xs:unique>
1470 </xs:element>
1471 <!-- -->
1472 <!-- Main type definitions -->
1473 <!-- -->
1474 <!-- The whole shebang -->
1475 <xs:complexType name="competencyDAGGraphType">
1476   <xs:sequence>
1477     <xs:element name="mapId" type="xs:anyURI" nillable="false"/>
1478     <xs:element ref="srcm:rcdRef" minOccurs="0"/>
1479     <xs:element ref="srcm:classLabel" minOccurs="0"/>
1480     <xs:element ref="srcm:title" minOccurs="0"/>
1481     <xs:element ref="srcm:description" minOccurs="0"/>
1482     <xs:element name="schemaLabel" type="xs:string" default="SRCM0.1.0"/>
1483     <xs:element name="schemaVersion" type="xs:string" default="1.0"/>
1484     <xs:element name="referential" type="xs:boolean" default="false" minOccurs="0"/>
1485     <xs:element name="metadata" type="srcm:metadataType" minOccurs="0"/>
1486     <xs:element name="graph" type="srcm:graphType">
1487       <xs:key name="KeyNodeIdentifier">
1488         <xs:selector xpath="//srcm:node"/>
1489         <xs:field xpath="@nodeId"/>
1490       </xs:key>
1491     </xs:element>

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1492     <xs:group ref="srcm:groupAny"/>
1493   </xs:sequence>
1494   <xs:attributeGroup ref="srcm:attGroupAny"/>
1495 </xs:complexType>
1496 <xs:complexType name="graphType">
1497   <xs:annotation>
1498     <xs:documentation>
1499       The DAG shaped graph, as a flat list of nodes.
1500       No ordering is implied, other than the directionality
1501       of the graph determined by parent to child associations..
1502     </xs:documentation>
1503   </xs:annotation>
1504   <xs:sequence>
1505     <xs:element name="defaultEntryNode">
1506       <xs:complexType>
1507         <xs:attribute name="nodeRef" type="srcm:localIdentifierType" use="required"/>
1508       </xs:complexType>
1509     </xs:element>
1510     <xs:element name="entryNodes">
1511       <xs:complexType>
1512         <xs:sequence>
1513           <xs:element name="entryNode" maxOccurs="unbounded">
1514             <xs:complexType>
1515               <xs:attribute name="nodeRef" type="srcm:localIdentifierType"
1516                 use="required"/>
1517             </xs:complexType>
1518           </xs:element>
1519         </xs:sequence>
1520       </xs:complexType>
1521     </xs:element>
1522     <xs:element name="node" type="srcm:nodeType" maxOccurs="unbounded"/>
1523     <xs:group ref="srcm:groupAny"/>
1524   </xs:sequence>
1525   <xs:attributeGroup ref="srcm:attGroupAny"/>
1526 </xs:complexType>
1527 <xs:complexType name="nodeType">
1528   <xs:annotation>
1529     <xs:documentation>
1530       Any node in the graph.
1531     </xs:documentation>
1532   </xs:annotation>
1533   <xs:sequence>
1534     <xs:element ref="srcm:classLabel" minOccurs="0"/>
1535     <xs:element ref="srcm:rcdRef" minOccurs="0"/>
1536     <xs:element ref="srcm:title" minOccurs="0"/>
1537     <xs:element ref="srcm:description" minOccurs="0"/>
1538     <xs:element name="metadata" type="srcm:metadataType" minOccurs="0"/>
1539     <xs:element name="parents" type="srcm:parentsType" minOccurs="0"/>
1540     <xs:element name="children" type="srcm:childrenType" minOccurs="0"/>
1541     <xs:element name="symLink" type="xs:anyURI" minOccurs="0"/>
1542     <xs:element name="rules" type="srcm:rulesType" minOccurs="0"/>
1543     <xs:group ref="srcm:groupAny"/>
1544   </xs:sequence>
1545   <xs:attribute name="nodeId" type="srcm:localIdentifierType" use="required"/>
1546   <xs:attributeGroup ref="srcm:attGroupAny"/>
1547 </xs:complexType>
1548 <!-- -->
1549 <!-- Various utility types used in the main types -->
1550 <!-- -->
1551 <xs:complexType name="classLabelType">
1552   <xs:annotation>
1553     <xs:documentation>
1554       Multilingual label with optional attributes to specify a model
1555       and token. The model can identify a modeling method or a vocabulary.
1556       The token can contain a language-independent name for the class.
1557     </xs:documentation>
1558   </xs:annotation>
1559   <xs:complexContent>
1560     <xs:extension base="srcm:langStringBagType">
1561       <xs:attribute name="model" type="xs:anyURI" use="optional"/>
1562       <xs:attribute name="token" type="xs:token" use="optional"/>

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1563     </xs:extension>
1564   </xs:complexContent>
1565 </xs:complexType>
1566 <xs:complexType name="rcdRefType">
1567   <xs:annotation>
1568     <xs:documentation>
1569       Reference to a rcd (typically a URN or handle) .
1570     </xs:documentation>
1571   </xs:annotation>
1572   <xs:sequence>
1573     <xs:group ref="srcm:groupAny"/>
1574   </xs:sequence>
1575   <xs:attribute name="ref" type="xs:anyURI"/>
1576   <xs:attributeGroup ref="srcm:attGroupAny"/>
1577 </xs:complexType>
1578 <xs:complexType name="parentsType">
1579   <xs:annotation>
1580     <xs:documentation>
1581       Collection of associations with parent nodes. No order is implied.
1582       These are empty elements but an application may use
1583       the extension facility to include additional elements inside.
1584     </xs:documentation>
1585   </xs:annotation>
1586   <xs:sequence>
1587     <xs:element name="parent" minOccurs="0" maxOccurs="unbounded">
1588       <xs:complexType>
1589         <xs:attribute name="nodeRef" type="srcm:localIdentifierType" use="required"/>
1590       </xs:complexType>
1591     </xs:element>
1592     <xs:group ref="srcm:groupAny"/>
1593   </xs:sequence>
1594   <xs:attributeGroup ref="srcm:attGroupAny"/>
1595 </xs:complexType>
1596 <xs:complexType name="childrenType">
1597   <xs:annotation>
1598     <xs:documentation>
1599       Collection of associations with child nodes. Application profiles
1600       determine whether those should be considered ordered or not.
1601       The elements include association qualifiers.
1602       The extension facility may be used to add application-specific
1603       qualifiers for the association.
1604     </xs:documentation>
1605   </xs:annotation>
1606   <xs:sequence>
1607     <xs:element name="child" minOccurs="0" maxOccurs="unbounded">
1608       <xs:complexType>
1609         <xs:sequence>
1610           <xs:element name="weight" type="srcm:weightType" default="1.0"
1611 minOccurs="0"/>
1612           <xs:element name="dataRequired" type="xs:boolean" default="true"
1613 minOccurs="0"/>
1614           <xs:element name="proficiencyRequired" type="srcm:proficiencyScoreType"
1615 minOccurs="0"/>
1616           <xs:element name="proficiencyDesired" type="srcm:proficiencyScoreType"
1617 minOccurs="0"/>
1618           <xs:group ref="srcm:groupAny"/>
1619         </xs:sequence>
1620         <xs:attribute name="nodeRef" type="srcm:localIdentifierType" use="required"/>
1621         <xs:attributeGroup ref="srcm:attGroupAny"/>
1622       </xs:complexType>
1623     </xs:element>
1624     <xs:group ref="srcm:groupAny"/>
1625   </xs:sequence>
1626   <xs:attributeGroup ref="srcm:attGroupAny"/>
1627 </xs:complexType>
1628 <xs:complexType name="rulesType">
1629   <xs:annotation>
1630     <xs:documentation>
1631       Rules associated directly with a node.
1632       See document.
1633     </xs:documentation>

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1634     </xs:annotation>
1635     <xs:sequence>
1636       <xs:element name="proficiencyRequired" type="srcm:proficiencyScoreType"
1637 minOccurs="0"/>
1638       <xs:element name="proficiencyDesired" type="srcm:proficiencyScoreType"
1639 minOccurs="0"/>
1640       <xs:element name="rollupMethod" type="srcm:rollupMethodType" default="all"
1641 minOccurs="0"/>
1642       <xs:element name="rollupParam" type="xs:string" minOccurs="0"/>
1643       <xs:group ref="srcm:groupAny"/>
1644     </xs:sequence>
1645     <xs:attributeGroup ref="srcm:attGroupAny"/>
1646 </xs:complexType>
1647 <!-- -->
1648 <!-- Generic type definitions usable in various schemas -->
1649 <!-- -->
1650 <xs:simpleType name="localIdentifierType">
1651   <xs:annotation>
1652     <xs:documentation xml:lang="en">
1653       URI rather than id to allow custom identifier syntax not
1654       restricted to NCName syntax.
1655     </xs:documentation>
1656   </xs:annotation>
1657   <xs:restriction base="xs:anyURI"/>
1658 </xs:simpleType>
1659 <xs:complexType name="metadataType" abstract="false">
1660   <xs:annotation>
1661     <xs:documentation xml:lang="en">
1662       Placeholder for zero or more metadata instances of any type.
1663       Also provides a basic name-value pair collection for named
1664       langString values. If the language feature is not required,
1665       the language attribute may be omitted as in
1666       (meta tag="my name")
1667         (langString)the string(/langString)
1668       (/meta)
1669     </xs:documentation>
1670   </xs:annotation>
1671   <xs:sequence>
1672     <xs:element name="meta" minOccurs="0" maxOccurs="unbounded">
1673       <xs:complexType>
1674         <xs:complexContent>
1675           <xs:extension base="srcm:langStringBagType">
1676             <xs:attribute name="tag" type="xs:string" use="required"/>
1677           </xs:extension>
1678         </xs:complexContent>
1679       </xs:complexType>
1680       <xs:unique name="UniqueMetaLanguageInstance">
1681         <xs:selector xpath="./langString"/>
1682         <xs:field xpath="@lang"/>
1683       </xs:unique>
1684     </xs:element>
1685     <xs:group ref="srcm:groupAny"/>
1686   </xs:sequence>
1687   <xs:attributeGroup ref="srcm:attGroupAny"/>
1688 </xs:complexType>
1689 <xs:simpleType name="real7Type">
1690   <xs:annotation>
1691     <xs:documentation xml:lang="en">
1692       As explained in IEEE 1484.11.1-2004, Annex B.1 Real data type
1693     </xs:documentation>
1694   </xs:annotation>
1695   <xs:restriction base="xs:decimal"/>
1696 </xs:simpleType>
1697 <xs:complexType name="proficiencyScoreType">
1698   <xs:annotation>
1699     <xs:documentation xml:lang="en">
1700       Can be used in 3 ways:
1701       a. As empty element with the scaled attribute set to a scaled score
1702          conformant to IEEE 1484.11.1
1703       b. Without the scaled attribute but with additional application-
1704          specific elements specifying an application specific

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1705         scale and value data.
1706     c. Combination of scaled value attribute and application-
1707         specific scale and value data.
1708     </xs:documentation>
1709 </xs:annotation>
1710 </xs:sequence>
1711     <xs:group ref="srcm:groupAny"/>
1712 </xs:sequence>
1713     <xs:attribute name="scaled" type="srcm:scaledScoreType" use="optional"/>
1714     <xs:attributeGroup ref="srcm:attGroupAny"/>
1715 </xs:complexType>
1716 <xs:simpleType name="scaledScoreType">
1717     <xs:annotation>
1718         <xs:documentation xml:lang="en">
1719             As defined in IEEE 1484.11.1-2004
1720         </xs:documentation>
1721     </xs:annotation>
1722     <xs:restriction base="srcm:real7Type">
1723         <xs:minInclusive value="-1"/>
1724         <xs:maxInclusive value="1"/>
1725     </xs:restriction>
1726 </xs:simpleType>
1727 <xs:simpleType name="weightType">
1728     <xs:annotation>
1729         <xs:documentation xml:lang="en">
1730             Weight as a fuzzy logic multiplier (0..1)
1731         </xs:documentation>
1732     </xs:annotation>
1733     <xs:restriction base="srcm:real7Type">
1734         <xs:minInclusive value="0"/>
1735         <xs:maxInclusive value="1"/>
1736     </xs:restriction>
1737 </xs:simpleType>
1738 <!-- -->
1739 <!-- Internationalization support -->
1740 <!-- -->
1741 <xs:complexType name="langStringBagType">
1742     <xs:annotation>
1743         <xs:documentation xml:lang="en">
1744             Unordered collection of 0 or more strings that should carry
1745             the same meaning in different languages. Each string
1746             must be tagged with a different lang attribute.
1747             One string may omit the lang attribute.
1748         </xs:documentation>
1749     </xs:annotation>
1750     <xs:sequence>
1751         <xs:element name="langString" type="srcm:langStringType"
1752             minOccurs="0" maxOccurs="unbounded"/>
1753     </xs:sequence>
1754 </xs:complexType>
1755 <xs:complexType name="langStringType">
1756     <xs:annotation>
1757         <xs:documentation xml:lang="en">
1758             String element with an optional language attribute.
1759             Application-specific additional attributes are allowed.
1760             See proposed standard document text.
1761         </xs:documentation>
1762     </xs:annotation>
1763     <xs:simpleContent>
1764         <xs:extension base="xs:string">
1765             <xs:attribute name="lang" type="xs:string" default=""/>
1766             <xs:attributeGroup ref="srcm:attGroupAny"/>
1767         </xs:extension>
1768     </xs:simpleContent>
1769 </xs:complexType>
1770 <!-- -->
1771 <!-- Simple types -->
1772 <!-- -->
1773 <!-- -->
1774 <!-- Attribute type definitions -->
1775 <!-- -->

```

This is an unapproved proposal for a Standards Draft, subject to change.

```

1776 <xs:simpleType name="rollupMethodType">
1777   <xs:annotation>
1778     <xs:documentation>
1779       Rollup method applied by a parent node when
1780       rolling up data from its children. See text.
1781     </xs:documentation>
1782   </xs:annotation>
1783   <xs:restriction base="xs:token">
1784     <xs:enumeration value="all"/>
1785     <xs:enumeration value="any"/>
1786     <xs:enumeration value="fraction"/>
1787     <xs:enumeration value="units"/>
1788     <xs:enumeration value="mean"/>
1789     <xs:enumeration value="other"/>
1790   </xs:restriction>
1791 </xs:simpleType>
1792 <!-- -->
1793 <!-- Extensibility support -->
1794 <!-- -->
1795 <xs:group name="groupAny">
1796   <xs:annotation>
1797     <xs:documentation>
1798       Any namespaced element from any namespace may be included
1799       where this placeholder appears. The namespace for the imported element
1800       must be defined in the instance, and the schema should be imported.
1801     </xs:documentation>
1802   </xs:annotation>
1803   <xs:sequence>
1804     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1805     maxOccurs="unbounded"/>
1806   </xs:sequence>
1807 </xs:group>
1808 <xs:attributeGroup name="attGroupAny">
1809   <xs:annotation>
1810     <xs:documentation>
1811       Any namespaced attribute from any namespace may be included
1812       where this placeholder appears. The namespace for the imported element
1813       must be defined in the instance, and the schema should be imported.
1814     </xs:documentation>
1815   </xs:annotation>
1816   <xs:anyAttribute namespace="##other" processContents="lax"/>
1817 </xs:attributeGroup>
1818 </xs:schema>

```