Conceptual framework for representing competency information

Version	Date	Comment
1.0	6 March 2013	First draft
1.1	28 June 2013	Updated diagram on page 2. Changes to key terms ("expression" instead of "measure", "representation" instead of "record", "attribution" and "attributor" instead of "claim" and "claimant"). New section 3 on the nature of competency. Definitions added as new section 9.

I give my thanks to Simon Grant of JISC/CETIS for his contributions to this document.

1 Introduction

This paper presents a "conceptual framework". This term is used to mean a set of interrelated terms, along with corresponding definitions and explanations of the relationships between them.

The terms have been selected with the intention that they can be used comfortably and without causing confusion. What matters is not the "true" meaning of any term, but rather the way in which the terms work together to provide a useful way of conceiving what has been a particularly challenging problem.

This conceptual framework can be implemented by software applications, software systems and published specifications, all of which at the moment are likely to use different information models. In time, the agreement of a single conceptual model may lead to the agreement of standardised data models.

Implementations are deemed to be conformant to this specification if they can map their information models to the terminology used in this conceptual model, while meeting all its provisions. Implementations NEED NOT adopt the exact terminology that is presented in the conceptual model.

2 Overview

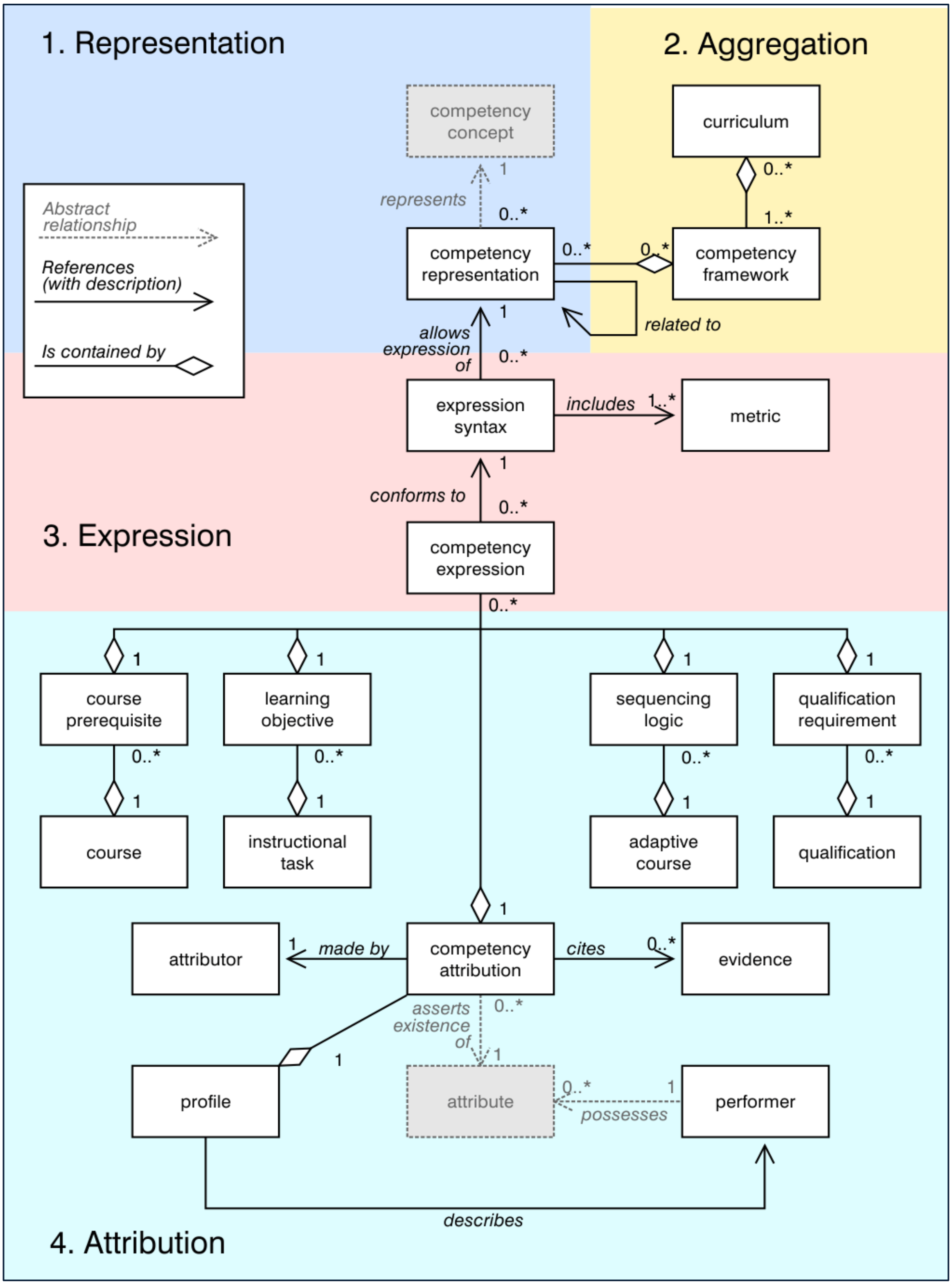

Figure 1

Figure 1 illustrates the key components of the conceptual framework and the relationships between them.

3 Competency

Competency itself is loosely defined as the ability, skill, knowledge, attitude, preference, other characteristic or combination of such characteristics, the possession of which will determine the extent and likelihood that a performer will achieve a certain type of outcome in a certain type of situation.

It is worth noting the following.

“Competency” is used as an abstract term to cover many different types of characteristic. This framework is not concerned to draw distinctions between knowledge, skill, ability or attitude: these are questions for psychologists, neurologists and other types of behavioural scientist. It will often be the case that we do not know exactly why or how someone can reliably perform in a particular way—we can only observe that he or she does so.

Competency is predictive of performance. If you say that someone is “good at math”, you mean that they *will* be able to solve certain sorts of problem in the future when required, not that they *did* solve certain sorts of problem in the past. Of course, the fact that they did solve certain problems in the past may be taken as evidence that they will be able to do so again in the future.

Competency is not necessarily tied to a single sort of performance. Statements like “she is a born leader” or “he is good at human relations” are predictive of a wide range of different sorts of performance. The astronauts who went to the moon were nearly all fighter pilots: at the time that they acquired their competencies, the context in which those competencies would ultimately be exercised had not yet been invented. It follows that competency is often abstract and transferable. This does not mean that other competencies may not be associated closely with particular tasks: the ability to play *Für Elise* on the piano is easily demonstrated by playing *Für Elise* on the piano. Being good at music is an abstract competency; the ability to play *Für Elise* on the piano is a concrete competency.

As competencies become more abstract and transferable, so people are likely to diverge in their interpretations of what a competency means. While people are unlikely to disagree very widely on the quality of a rendition of *Für Elise*, people may have very different views of what constitutes good leadership. This does not mean that we should only measure concrete competencies that are tightly coupled to particular, well-defined performances: it is in the possession of abstract, transferable competencies that the true value of education lies.

Competency does not refer to capabilities that are conferred by external circumstances, such as the possession of equipment, a favourable situation, or the support of other people.

Learning may properly be defined as an improvement in competency. However, in common use it is often defined as the possession of a degree of a particular competency—corresponding in other words to an absolute competency level and not to a competency delta.

4 Competency concept

A competency concept is intangible, may be poorly defined, and is the only object included in Figure 1 that cannot correspond to an actual data structure. It represents an idea of competency that might be shared between many different people, organisations and cultures.

Until a competency concept is represented more precisely, there is significant scope for misunderstanding in the way that different people perceive different competency concepts.

5 Competency representation

5.1 Overview

A competency representation is a data structure that identifies and describes a competency concept, providing an actual data structure to represent what is otherwise intangible and poorly defined.

There MAY be many competency representations, created by different authorities and authors, that represent the same or similar competency concepts.

A competency representation SHALL contain:

- a globally unique identifier that CAN be used to reference the representation globally;
- at least one human readable descriptor for the competency concept that it describes.

A competency representation MAY contain:

- links to external information or resources;
- information about its relationships to other competency representations;
- a formally structured competency descriptor.

5.2 Descriptors

The term “descriptor” is applied to any human-readable property that that describes or informally identifies the competency concept being expressed. Different descriptors may be subject to different rules and constraints (such as restricted character sets, minimum or maximum lengths etc.). Typically, descriptors may be called:

- title;
- abbreviation;
- description.

Being human-readable, descriptors SHALL be marked up, individually or collectively, with the language in which they are expressed.

NOTE. Globally unique identifiers are not classed as descriptors on the grounds that they are designed to be machine-readable.

5.3 Links to external information and resources

The descriptors contained within a competency representation are unlikely to be sufficient to ensure the consistent understanding and application of a competency concept throughout a community of practice. In most circumstances, further elucidation may be required.

For this reason, the competency representation does not "define", but rather "identifies" and "describes" a competency concept.

The following are examples of different types of external documentary support that might be provided:

- documents that provide extensive explanations of the concept described;
- regular reports that update and clarify the community's understanding of the concepts described, often in response to examinations or other evidence of practice;
- examples and use cases that illustrate the application of the concept.

The following are examples of other resources that might be provided:

- online forums in which practitioners might discuss the application of the concept in different circumstances;
- access to expert advice and help-lines.

5.4 Formally structured competency descriptor

Formally structured competency descriptors convey the meaning of a competency concept using data that conforms to a formal data model.

EXAMPLE. A structured competency description might use a prescribed grammar, such as "subject—verb—action", providing rules and vocabularies to specify how such statements can be constructed.

6 Competency frameworks and relationships

6.1 Overview

One competency is frequently held to be associated with or dependent on another competency. These associations are modelled in implementations by the use of frameworks and relationships.

This conceptual framework does not prescribe how such relationships are encoded. It merely provides a way of classifying different approaches to competency frameworks and relationships as a means of encouraging the emergence of a consensus about how these should be standardised.

6.2 Relationships between competency representations

Competency frameworks MAY support different kinds of relationships between competency representations.

For the purposes of this specification:

- all relationships SHALL involve two separate competency representations;
- all relationships SHALL be declared to be of a particular type.

Competency frameworks that support relationships between competency representations SHALL provide a list of the types of relationship that are available for use, along with any rules that govern their use.

Competency frameworks that support relationships between competency representations NEED NOT support more than one type of relationship.

EXAMPLE. Typical relationships might include:

- *x is equivalent to y;*
- *x is broader than y;*
- *x is narrower than y;*
- *x is similar to y.*

EXAMPLE. An implementation may specify that a competency representation may only be involved in one relationship of type "is narrower than".

6.3 Competency frameworks

Competency frameworks are collections of competency representations.

Implementations MAY support competency frameworks.

NOTE. Different implementations often use different synonyms for "competency framework", such as "competency structure", "competency organization", "competency model" or "competency network".

Implementations that support competency frameworks SHALL specify:

- either that a competency representation is permitted to belong to more than one framework;
- or that a competency representation is permitted to belong to only one framework.

Where implementations support frameworks, those frameworks:

- SHALL contain a unique identifier that CAN be used to reference the framework globally;
- SHALL contain at least one human readable descriptor;
- SHALL contain information specifying which competency representations are members of the framework;

- MAY contain information about relationships between competency representations;
- MAY contain information in addition to that provided for above.

Where a framework contains information on relationships between competency representations, at least one of the two competency representations involved in each relationship SHALL be a member of the framework that contains this information.

Implementations that support frameworks MAY support different categories of membership and MAY provide rules for the use of those different categories of membership.

EXAMPLE. A competency representation may belong to one "default" or "home" framework and any number of "supplementary" frameworks.

6.4 The relationships of a competency representation

An implementation MAY provide rules that allow:

- a competency representation to contain information about its relationships with other competency representations;
- a competency representation to contain information about the frameworks to which it belongs.

7 Expression syntax

7.1 Overview

An expression syntax specifies how formally to express competency in respect of a particular competency representation.

EXAMPLE. A competency representation might represent the ability to type. A competency expression might express the ability to type at 35 words per minute and it is the competency expression (not the competency representation) that can be attributed to an individual.

NOTE. In some cases an implementation may wish to do no more than to attribute to an individual the ability to type. In this case, the competency expression and the expression syntax are still used, if only for the sake of consistency. In this case the expression syntax would be of type "Boolean" and the competency expression would be "true".

7.2 Metrics

7.2.1 Overview

An expression syntax SHALL be composed of one or more metrics.

Implementations NEED NOT allow more than one metric to be included in each expression syntax.

Where implementations allow for more than one metric, each metric SHALL:

- contain at least one descriptor;
- be uniquely identifiable within the local syntax.

EXAMPLE. An implementation might require each metric to have a "label" property, where each value for "label" was required to be unique among other metrics belonging to the same syntax.

Each metric expresses one aspect of the associated competency. Metrics MAY represent:

- different ways in which a performance might be assessed;
- boundaries to the range of performances that might be expected;
- environmental conditions that would affect the level of performance to be expected.

EXAMPLE. A typing competency designed to be expressed with a high degree of precision might use the 6 metrics shown in Table 1.

Metric descriptor	**Type**	**Comment**
Words Per Minute	Integer	
Errors Per Minute	Integer	
Language complexity	Integer (1–10)	
Working conditions	Enumerated	Poor, Fair, Satisfactory, Good or Excellent.
Keyboard	Enumerated	Vocabulary provided, e.g. "en-UK"
Rating	Integer (0-100)	Derived from other metrics

Table 1

An expression syntax for typing, designed for a low degree of precision might use the 11 metrics shown in Table 2.

Metric descriptor	**Type**	**Comment**
Words Per Minute (Low)	Integer	
Words Per Minute (High)	Integer	
Errors Per Minute (Low)	Integer	
Errors Per Minute (High)	Integer	
Language complexity (Low)	Integer (1–10)	
Language complexity (High)	Integer (1–10)	

Working conditions (Low)	Enumerated	Scale of Poor, Fair, Satisfactory, Good, Excellent.
Working conditions (High)	Enumerated	
Keyboard	Enumerated	
Calculated Rating (Low)	Integer (0-100)	Calculated from other metrics
Calculated Rating (Low)	Integer (0-100)	Calculated from other metrics

Table 2

7.2.2 Metric types

Each metric SHALL be of one of the following types:

- complex;
- Boolean;
- integer;
- decimal;
- enumerated.

Implementations SHALL support at least one of these types but NEED NOT support them all.

Implementations MAY provide rules for the declaration of metrics whose inclusion in any particular competency expression may be specified as optional, mandatory, or conditional.

7.2.3 Complex metrics

A complex metric expresses competency using data that conforms to a formal specification.

Implementations that support complex metrics SHALL provide a set of rules that specify how such expressions should be constructed and interpreted.

7.2.4 Boolean metric

A Boolean metric expresses competency as one of two levels:

- true (maximum competency);
- false (no competency).

7.2.5 Integer metric

An integer metric expresses competency as a signed (positive or negative) whole number. Implementations MAY allow integer metrics to be restricted to being:

- more than a minimum value;
- less than a maximum value;
- more or equal to a minimum value;

- less or equal to a maximum value.

NOTE. A numeric metric (integer or decimal) that was constrained to have a minimum value of 0 would be equivalent to an unsigned (positive only) number.

7.2.6 Decimal metric

A decimal metric expresses competency as a floating point number. Implementations MAY allow decimal metrics to be restricted:

- to being more than a minimum value;
- to being less than a maximum value;
- to a specified number of decimal places;
- more or equal to a minimum value;
- less or equal to a maximum value.

EXAMPLE. A syntax designed to express "speed at sprinting 100 metres" might specify a decimal metric for "seconds", to be expressed to two decimal places.

7.2.7 Enumerated metric

An enumerated metric provides a vocabulary by which different levels of mastery can be indicated (see example of "working conditions" given in Table 1).

7.3 Placement of expression syntax

7.3.1 Overview

Implementations MAY take different approaches to the incorporation of expression syntaxes. These alternative approaches are reflected in the different placement of the expression syntax within their different information models.

Implementations SHALL use one of the following placements:

- explicit expression syntax;
- embedded expression syntax;
- implicit expression syntax;
- no expression syntax.

7.3.2 Explicit expression syntaxes

Explicit expression syntaxes are declared as separate data structures.

An explicit expression syntax SHALL contain:

- a globally unique identifier that CAN be referenced globally;
- at least one descriptor.

In addition to meeting general provisions for all types of expression syntax, implementations that use explicit expression syntaxes:

- SHALL specify how to declare an expression syntax as a separate data structure;
- SHALL allow at least two expression syntaxes to be associated with a single competency representation;
- SHOULD provide a mechanism for mapping expressions that conform to the different syntaxes that reference the same representation.

EXAMPLE. A typing competency might be expressed according to one of two syntaxes: one specifying words per minute; and one specifying an enumerated metric with a vocabulary of “beginner”, “intermediate” or “advanced”. The implementation should provide a mechanism to allow a competency expressed in terms of words per minute to be mapped to a competency expressed in terms of levels, as illustrated by Table 3 below.

Syntax 1 metric descriptor	**Mapping**	**Syntax 2 metric descriptor**
“words per minute”		*“level”*
Integer	From 0 WPM	Beginner
	To 15 WPM	
	From 16 WPM	Intermediate
	To 30 WPM	
	From 31 WPM	Advanced
	No limit	

Table 3

Mapping mechanisms NEED NOT be lossless.

EXAMPLE. A competency that was expressed as “18 words per minute” might be mapped to the “intermediate” level. If it were then mapped back to “words per minute”, it could only be translated to “between 16 and 30 words per minute". Precision would have been lost.

7.3.3 Embedded expression syntaxes

An embedded expression syntax refers to the circumstance where no separate expression syntax data structure is provided, but where competency representations themselves contain expression syntax information.

Implementations that use embedded expression syntaxes:

- SHALL meet all the provisions of this document for expression syntaxes in the rules provided for declaring competency representations;
- SHALL NOT allow more than one expression syntax to be associated with a single competency representation.

7.3.4 Implicit expression syntaxes

Implicit expression syntaxes refer to implementations where the rules for expressing competency are fixed, being contained within the specification itself and being applied equally to all competency representations.

Implementations that use implicit expression syntaxes SHALL provide a single set of rules for the expression of competency that SHALL apply to all competency representations.

EXAMPLE. An implementation may specify that all competency is to be expressed as a Boolean value.

7.3.5 No expression syntaxes

Implementations that use no expression syntaxes SHALL NOT allow for the expression or attribution of competency in any form.

8 Competency expression

8.1 Overview

A competency expression contains data that specifies a particular type or degree of competency, following the rules specified by the expression syntax to which it conforms.

In implementations that use explicit expression syntaxes, a competency expression SHALL contain a reference to the expression syntax according to which the data is formatted.

In implementations that use embedded or implicit expression syntaxes, a competency expression SHALL contain a reference to the competency representation to which the expression refers.

The following table provides examples of competency expressions, with details of the expression syntaxes and the titles of competency representations that they are associated with. All of the expression syntaxes in the table below have single metrics.

Expressed as…	**…using metric…**		**…for competency representation**
Expression	*Type*	*Label*	*Title*
35	Integer	Words per minute	Typing
11.25	Decimal	Seconds	Sprinting over 100m
Intermediate	Enumerated	Level	Skiing
True	Boolean	Status	Health & safety training

Competency expressions may be used in many different circumstances, including in:

- competency attributions;
- learning objectives;
- course prerequisites;
- adaptive logic statement;
- qualification requirements.

8.2 Competency attribution

A competency attribution asserts that a particular performer has mastered a particular competency in a specified way or to a specified degree.

Implementations MAY support competency attributions.

In implementations that support competency attributions, each competency attribution:

- SHALL contain a competency expression;
- SHALL EITHER contain a reference to a performer OR SHALL be contained within another data structure that contains a reference to a performer;
- MAY be contained within a data structure containing general student profile information.

Additionally, each competency attribution MAY contain:

- a reference to a attributor;
- an indication of the confidence with which the attribution is asserted;
- evidence that supports the attribution (either copied into the attribution representation or referenced by the attribution representation).

8.3 Learning objective

A learning objective indicates an intended outcome of an instructional process or task. Learning objectives are a feature of formal education, in which activities are planned and may often be assigned to students.

One or more learning objectives are typically contained within a data structure that represents a learning activity, course, programme of study or other instructional process.

Implementations MAY support learning objectives. If they do, each learning objective SHALL contain a competency expression.

NOTE. In this context, the word "learning" is not used with precision, as it refers to the possession of an absolute degree of competency and not to a relative improvement in competency. It is entirely feasible that students might satisfy a learning objective at the end of a course, even though, having been able to satisfy the learning objective at the beginning of the course, they experienced no learning whatsoever during the course. While this lack of precision is regrettable, *the term "learning objective" is nevertheless used on account of its currency. The conceptual difficulties presented by its use can be mitigated by appropriate use of learning prerequisites.*

8.4 Learning prerequisites

A learning prerequisite indicates a competency expression which a student is either required or advised to have either attained or not attained before engaging in an instructional process. Learning prerequisites are a feature of formal education, in which activities are planned and progression is actively managed.

One or more learning prerequisites are typically contained within a data object that represents a learning activity, course, programme of study or other instructional process.

Implementations MAY support learning prerequisites. If they do, each learning prerequisite SHALL contain a competency expression.

8.5 Adaptive logical statement

There is increasing interest in systems to support adaptive learning. Such systems normally contain logical statements that manage students' progression or otherwise adapt the behaviour of a learning activity or process. These adaptations typically depend on many different variables, one of which might be the extent and nature of a student's competency.

Implementations MAY support adaptive logic that references competency. If they do, each adaptive logic clause SHALL contain a competency expression, allowing adaptations to be made depending on whether particular participants have satisfied the expression or not.

8.6 Qualification criteria

Qualifications are typically awarded on the basis of a number of criteria. Some or all of these criteria may refer to the student having attained one or more degrees of competency that are associated with different competency expressions.

Qualification criteria are typically contained within a data structure representing a particular qualification.

Implementations MAY support qualification criteria, as described in this clause. If they do, each qualification criterion SHALL contain a competency expression, on the basis of which the qualification being described is to be awarded.

9 Definitions

actor

person, group of people, organisation or software system that is capable of acting.

performance

one or more actions completed with the intention of achieving a particular outcome.

performer

actor that has produced or is expected to produce at least one **performance**.

competency

ability, skill, knowledge, attitude, preference, other characteristic or combination of such characteristics, the mastery of which will determine the extent and likelihood that a **performer** will achieve a certain type of outcome in a certain type of situation.

NOTE. The term "competency" does not refer to capabilities that are conferred by external circumstances, such as the possession of equipment, a favourable situation, or the support of other people.

concept

intangible idea, lacking in formal definition.

NOTE. For example, a competency concept is the intangible idea of a certain sort of competency.

representation

data structure that identifies and describes a concept, process, or other non-digital object.

NOTE 1. For example, a competency representation identifies and describes a competency concept.

NOTE 2: There may be many different competency representations, with different identifiers, that describe similar competency concepts.

descriptor

human-readable, textual property designed to describe a concept, process or object.

NOTE 1. Common names for descriptors are "title", "description" or "abbreviation".

NOTE 2. Descriptors may also identify an entity or representation, especially within a local context.

formally structured competency description

machine-readable description of the meaning of a **competency concept**, contained within a **competency representation** and conforming to an external data specification.

expression syntax

data structure that provides rules to specify how a particular **competency representation** may be quantified and expressed.

metric

single data element contained in an **expression syntax**.

NOTE 1. Metrics are likely to consist of signed or unsigned integers, floating point, Boolean or enumerated values.

NOTE 2. An expression syntax may have more than one metric. For example, a syntax for measuring typing ability may include a "words per minute" metric and an "errors per minute" metric.

NOTE 3. Expression syntaxes provide machine readable, actionable data and do not include unstructured text.

level

vocabulary element used by an metric of type "enumerated".

expression

data that identifies a particular degree of **competency**, conforming to the rules specified by a particular **expression syntax**.

attribute

characteristic of a particular **performer**, corresponding to a particular **competency concept**.

NOTE. Although the competency attributes of a performer may be inferred by observing his or her performances, they cannot be observed directly or ever established or quantified with certainty.

attribution

assertion that a certain **performer** possesses an **attribute** in respect of a certain **competency representation**.

attributor

actor who makes an **attribution**.

claimant

attributor who makes an **attribution** in respect of which he or she is also the **performer**.

competency framework

collection of related **competency representations**.