Conceptual framework for representing competency information

Version	Date	Comment
1.0	6 March 2013	First draft

My thanks to contributions made to this document by Simon Grant of JISC/CETIS.

1   Introduction

This conceptual framework can be implemented by software applications, software systems or by published specifications. It is anticipated that all these different implementations will use or propose their own information models.

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

The conceptual framework comprises a number of different objects and specifies the relationships between them.
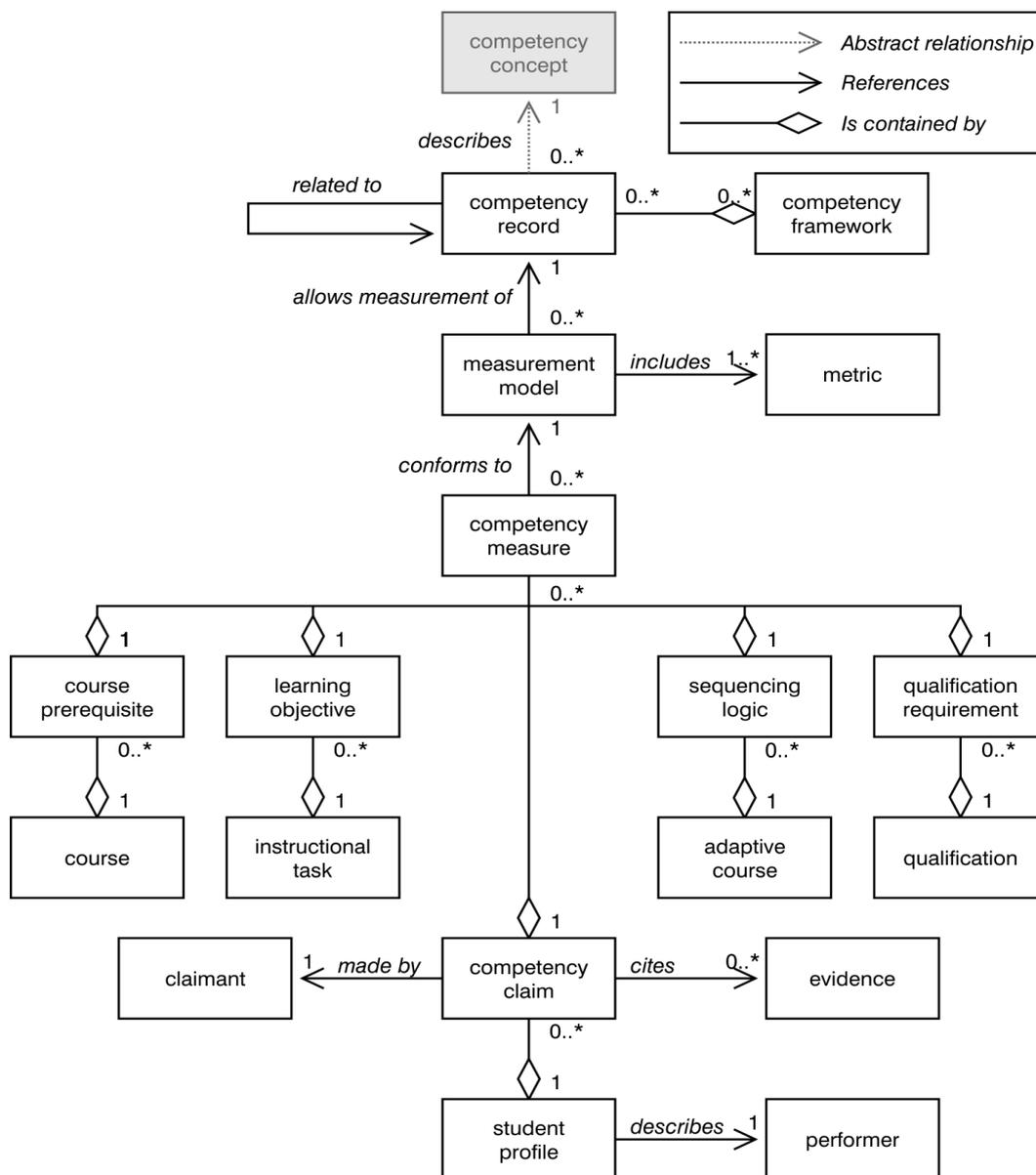
2   Overview

Figure 1

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

3   Competency concept

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

4   Competency record

4.1   Overview

A competency record 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 records, created by different authorities and authors, that represent the same or similar competency concepts.

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

A competency record MAY contain:
- links to external information or resources;
- information about its relationships to other competency records;
- a formally structured competency descriptor.

4.2   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.

4.3   Links to external information and resources

The descriptors contained within a competency record 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 record does not "define", but rather "identifies", "represents" 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.

4.4   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.

5   Competency frameworks and relationships

5.1   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.

5.2   Relationships between competency records

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

For the purposes of this specification:

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

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

EXAMPLE. Typical relationship 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 record may only be involved in one relationship of type "is narrower than".

5.3   Competency frameworks

Competency frameworks are collections of competency records.

Implementations MAY support competency frameworks.

Implementations that support competency frameworks SHALL specify:
- either that a competency record is permitted to belong to more than one framework;
- or that a competency record is permitted to belong to only one framework.

Where implementations support frameworks, those frameworks:
- SHALL contain a globally 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 records are members of the framework;
- MAY contain information about relationships between competency records;
- MAY contain information in addition to that provided for above.

Where a framework contains information on relationships between competency records, at least one of the two competency records 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 record may belong to one "default" or "home" framework and any number of "supplementary" frameworks.

5.4   The relationships of a competency record

An implementation MAY provide rules that allow:
- a competency record to contain information about its relationships with other competency records;
- a competency record to contain information about the frameworks to which it belongs.

6 Measurement models

6.1 Overview

A measurement model specifies how formally to express particular competency measures.

EXAMPLE. A competency record might represent the ability to type. A competency measure might represent the ability to type at 35 words per minute and could be attributed to an individual. In this case, a measurement model would contain the set of rules which specified how the competency record could be expressed as a measure.

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 measure and the measurement model are still used, if only for the sake of consistency. In this case the measurement model would be of type "Boolean" and the competency measure would be expressed as "true".

6.2 Measurement metrics

6.2.1 Overview

A measurement model SHALL be composed of one or more measurement metrics.

Implementations NEED NOT allow more than one measurement metric to be included in each measurement model.

Where implementations allow for more than one measurement metric, each metric SHALL:
- contain at least one descriptor;
- be uniquely identifiable within the local measurement model.

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

Each measurement metric expresses one aspect of the competency that is to be expressed. 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)	Calculated from other metrics

Table 1

A measurement model 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

6.2.2 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 measure may be specified as optional, mandatory, or conditional.

6.2.3   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.

6.2.4   Boolean metric

A Boolean metric expresses competency as one of two levels:
- true (maximum competency);
- false (no competency).

6.2.5   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;
- to being less than 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.

6.2.6   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.

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

6.2.7   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).

6.3   Placement of measurement model

6.3.1   Overview

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

Implementations SHALL use one of the following placements:
- explicit measurement models;
- embedded measurement models;
- implicit measurement models;
- no measurement models.

6.3.2   Explicit measurement models

Explicit measurement models are declared as separate data structures.

An explicit measurement model 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 measurement model, implementations that use explicit measurement models:

SHALL specify how to declare a measurement model as a separate data structure;

SHALL allow at least two measurement models to be associated with a single competency record;

SHOULD provide a mechanism for mapping expressions that conform to the different models that reference the same record.

EXAMPLE. A typing competency might be expressed according to one of two models: 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.

Model 1 metric descriptor	Mapping	Model 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.

6.3.3 Embedded measurement models

An embedded measurement model refers to the circumstance where no separate measurement model data structure is provided, but where competency records themselves contain measurement model information.

Implementations that use embedded measurement models:

SHALL meet all the provisions of this document for measurement models in the rules provided for declaring competency records;

SHALL NOT allow more than one measurement model to be associated with a single competency record.

6.3.4 Implicit measurement models

Implicit measurement models refer to implementations where the rules for expressing competency are fixed, being contained within the specification itself and being applied equally to all competency records.

Implementations that use implicit measurement models SHALL provide a single set of rules for the expression of competency that SHALL apply to all competency records.

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

6.3.5 No measurement models

Implementations that use no measurement models SHALL NOT allow for the expression of competency in any form.

7   Competency measure

7.1   Overview

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

In implementations that use explicit measurement models, a competency measure SHALL contain a reference to the measurement model according to which the data is formatted.

In implementations that use embedded or implicit measurement models, a competency measure SHALL contain a reference to the competency record to which the expression refers.

The following table provides examples of competency measures, with details of the measurement models and the titles of competency records that they are associated with. All of the measurement models in the table below have single metrics.

Measure of...	Measurement Model Metric		Competency record
Expression	*Type*	*Label*	*Title*
35	Integer	Words per minute	Typing
11.25	Decimal	Seconds	Sprinting over 100m
Intermediate	Enumerated	Level	Skiing
Mastered	Boolean	Status	Health & safety training

Competency expressions may be used in many different circumstances, including in:
- competency claims;
- learning objectives;
- course prerequisites;
- adaptive logic statement;
- qualification requirements.

7.2   Competency claim

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

Implementations MAY support competency claims.

In implementations that support competency claims, each competency claim:
- SHALL contain a competency measure;

- 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 claim MAY contain:
- a reference to a claimant;
- an indication of the confidence with which the claim is asserted.
- evidence that supports the claim (either copied into the claim record or referenced by the claim record).

7.3   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 measure.

7.4   Learning prerequisites

A learning prerequisite indicates a competency measure which a student is either required or advised to attain 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 measure.

7.5   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 measure, allowing

adaptations to be made depending on whether particular participants have satisfied the measure or not.

7.6   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 competency measures.

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 measure, on the basis of which the qualification being described is to be awarded.