



COMPER

**Une approche par compétences pour le diagnostic, la
régulation et la personnalisation de l'apprentissage**

Lot 1 : modélisation du référentiel de compétences

Livrable

Nom	Méta-modèle du référentiel de compétences fondé sur le web sémantique
Numéro	L1.1
Partenaire responsable	IRIT (J. Broisin, A. Bey), Educlever (L. Polese)
Date de soumission	Mars 2020

Table des matières

Table des matières	2
1. Introduction	3
2. Les objets de type Framework	3
3. Les objets de type Framework_Object	4
4. Les objets de type Competency, Skill et Knowledge	5
4.1. Les objets de type Competency	6
4.2. Les objets de type Knowledge	7
4.3. Les objets de type Skill	7
5. Les objets de type Resource	7
Références	8

1. Introduction

L'objectif du lot 1 "Modélisation du référentiel de compétences" est de proposer un méta-modèle de représentation des compétences (i) capable d'exprimer une compétence à différents niveaux de granularité, (ii) suffisamment générique pour fédérer les compétences induites par des disciplines variées et des activités pédagogiques hétérogènes telles que celles proposées par les 3 terrains d'expérimentation du projet, (iii) et offrant l'opportunité de regrouper différentes compétences dans une même entité afin de former un référentiel de compétences et ainsi pouvoir considérer différents référentiels. Ce modèle doit pouvoir s'associer d'une part au profil cognitif de l'apprenant pour signifier la manipulation d'une compétence par l'apprenant à un moment donné et exprimer son niveau de maîtrise, et d'autre part aux modèles d'activités pédagogiques afin de faire correspondre une ou plusieurs compétences à un type d'activité donné.

Ce rapport décrit le méta-modèle du référentiel de compétences proposé par les partenaires IRIT et Educlever après diverses discussions avec l'ensemble du consortium. Le modèle illustré par la Figure 1 s'appuie sur le modèle COCON de Educlever (Pelap et al., 2018), et propose de l'étendre afin d'y apporter davantage de sémantique et d'expressivité.

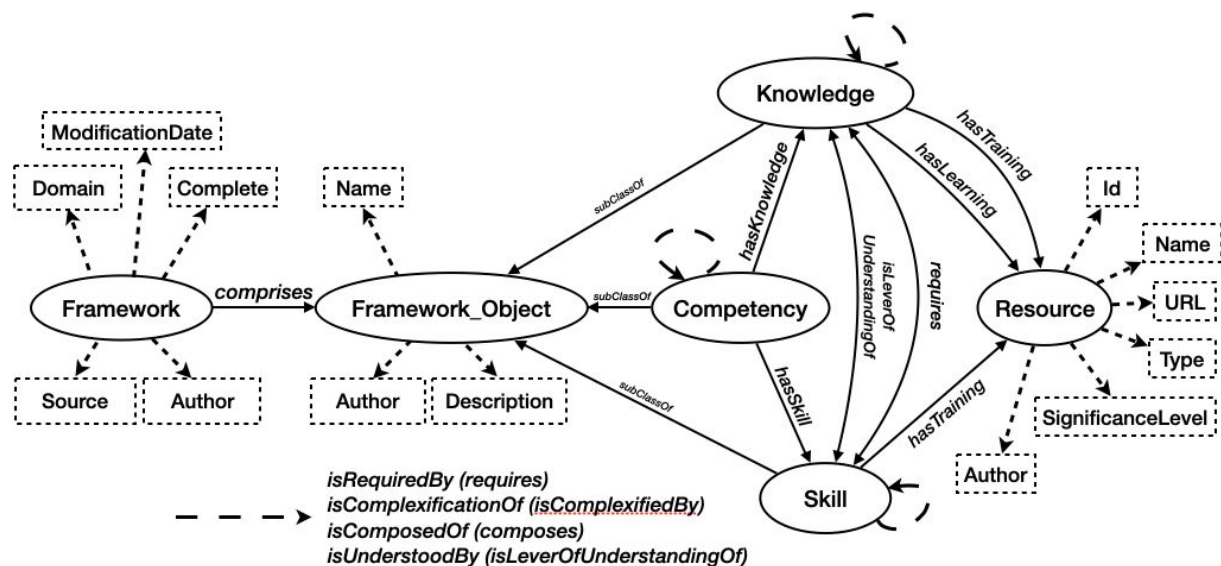


Figure 1. Le méta-modèle du référentiel de compétences

Dans la suite de ce rapport, nous définissons chaque nœud et relation de ce méta-modèle.

2. Les objets de type *Framework*

Un *Framework* (référentiel en français) permet de regrouper un ensemble de compétences, connaissances et savoir-faire au sein d'une même entité. Un *Framework* correspond par exemple à l'ensemble des compétences définies pour une discipline à un niveau d'apprentissage donné, ou aux compétences définies par une institution de standardisation pour un corps de métiers donné.

Un objet de type `Framework` est caractérisé par un ensemble de propriétés :

- **Domain** : cette propriété, exprimée sous la forme d'une chaîne de caractères, représente le domaine ou la discipline dans lequel s'inscrit le référentiel.
Exemples : Informatique ; Physique/Chimie ; Français.
- **Source** : cette propriété, exprimée sous la forme d'une chaîne de caractères, permet de définir la source du référentiel.
Exemples : Programme Pédagogique National DUT Informatique 2013 ; European e-Competence Framework (CEN, 2014).
- **Author** : cette propriété, exprimée sous la forme d'une chaîne de caractères, indique le ou les auteurs du référentiel. Un auteur peut être un individu ou une organisation.
Exemples : Jean Montanier ; Jean Montanier et Marcel Dupond ; IUT A Informatique, Université Paul Sabatier Toulouse 3.
- **ModificationDate** : cette propriété, exprimée sous la forme d'une chaîne de caractères, indique la date de dernière modification d'un référentiel. Cette date doit être exprimée au format yyyy/mm/dd.
Exemple : 2020/03/23.
- **Complete** : cette propriété, exprimée sous la forme d'une chaîne de caractères, indique si un référentiel est finalisé. Un référentiel dont cette propriété vaut "yes" peut être utilisé par une plate-forme externe.
Exemple : yes ; no.

3. Les objets de type *Framework_Object*

Un `Framework_Object` représente un élément renfermé dans un référentiel. Un tel objet peut être une compétence (*Competency*), un savoir-faire (*Skill*) ou une connaissance (*Knowledge*).

Un objet de type `Framework_Object` est caractérisé par un ensemble de propriétés :

- **Name** : cette propriété, représentée sous la forme d'une chaîne de caractères, désigne le nom donné à l'objet.
Exemples : Analyser-ANA ; Concevoir un algorithme récursif ; Manipuler des variables.
- **Description** : cette propriété, représentée sous la forme d'une chaîne de caractères, permet de décrire l'objet.
Exemples : dans le référentiel du C2i¹, la compétence "Pérenniser ses données" est décrite par "Mesurer les risques de perte, d'erreur et d'obsolescence de ses données et les préserver à court et à long terme".
- **Author** : cette propriété, représentée sous la forme d'une chaîne de caractères, indique le ou les auteurs de l'objet. Un auteur peut être un individu ou une organisation.

¹ <https://c2i.enseignementsup-recherche.gouv.fr/>

Exemples : Jean Durand ; Jean Durand et Marcel Dupond ; IUT A Informatique, Université Paul Sabatier Toulouse 3.

Un objet de type `Framework_Object` appartient à un et un seul référentiel. Ce lien d'appartenance est modélisé à travers la relation **isComprisedIn/comprised** avec un objet de type `Framework`. D'autre part, les relations qui peuvent être utilisées entre objets de type `Framework_Object` sont les suivantes :

- **isRequiredBy/requires** : cette relation exprime le fait qu'un objet est nécessaire à la compréhension d'un autre objet.
Exemples : la compétence "Manipuler des fichiers Unix" requiert la compétence "Exécuter des commandes Unix" ; la connaissance "Connaître la représentation d'une liste" est requise par la connaissance "Connaître les fonctions sur les listes".
- **isComplexificationOf/isComplexifiedBy** : cette relation exprime le fait qu'un objet est une complexification d'un autre objet. Cette relation implique forcément une relation de type `requires`, et dénote en plus le niveau de complexité qui peut exister entre deux objets de même type et traitant du même sujet.
Exemple : la compétence "Savoir additionner deux nombres" est une complexification de la compétence "Savoir additionner deux chiffres".
- **isComposedOf/composes** : cette relation est utilisée pour exprimer la spécialisation/généralisation/décomposition entre deux objets de même type.
Exemple : la compétence "Programmer en langage Shell" est composée, entre autres, des compétences "Manipuler des fichiers Unix" et "Exécuter des commandes Unix".
- **isUnderstoodBy/isLeverOfUnderstandingOf** : cette relation est utilisée pour exprimer le fait qu'un objet facilite la compréhension d'un autre objet de même type. Cette relation indique qu'un objet aide à comprendre un autre objet, alors que la relation `isRequiredBy/requires` décrite ci-dessus implique qu'un objet doit nécessairement être acquis pour être capable d'acquérir un autre objet.
Exemple : la compétence "Éditer des documents numériques avec Apache OpenOffice" aide à l'acquisition de la compétence "Éditer des documents numériques avec LibreOffice".

4. Les objets de type *Competency*, *Skill* et *Knowledge*

Les compétences représentent un concept complexe qui a été défini dans le cadre de nombreuses recherches. Il n'y a cependant pas de définition consensuelle et universelle de ce concept, mais la littérature montre que les compétences font souvent référence à un ensemble de connaissances, savoir-faire, capacités et autres attitudes et expériences personnelles (Campion et al., 2011; International Project Management Association, 2015; Kang and Ritzhaupt, 2015). A l'échelle européenne, la définition adoptée par le parlement européen le 26 septembre 2006 est la suivante :

« Une compétence est une combinaison de connaissances, d'aptitudes (capacités) et d'attitudes appropriées à une situation donnée. Les

compétences clés sont celles qui fondent l'épanouissement personnel, l'inclusion sociale, la citoyenneté active et l'emploi »

A l'échelle nationale, la définition retenue dans le socle commun de connaissances et de compétences reprend les concepts fondamentaux de la définition européenne :

« Chaque grande compétence du socle est conçue comme une combinaison de connaissances fondamentales pour notre temps, de capacités à les mettre en œuvre dans des situations variées mais aussi d'attitudes indispensables tout au long de la vie, comme l'ouverture aux autres, le goût pour la recherche de la vérité, le respect de soi et d'autrui, la curiosité et la créativité. »

Nous nous inspirons de ces définitions dans le cadre du projet COMPER, sans toutefois prendre en considération les attitudes citoyennes. En effet, l'objectif du projet COMPER est de faciliter l'acquisition de connaissances et capacités caractérisées par un faible niveau de granularité et nécessaires à la personnalisation des ressources proposées aux apprenants. Le projet n'a pas pour ambition de traiter des compétences généralistes, comportementales de haut niveau, même si le modèle que nous proposons offre la possibilité de définir des connaissances et capacités généralistes, tout en offrant des opportunités d'extension.

Les termes *compétence* et *capacité* sont parfois confondus. Dans le contexte du projet COMPER, nous adoptons les définitions proposées par Meirieu (1987, p. 153-154) : une capacité est une « *activité intellectuelle stabilisée et reproductible dans des champs divers de la connaissance* », alors qu'une compétence est définie comme « *un savoir identifié mettant en jeu une ou des capacités, dans un champ notionnel ou disciplinaire déterminé* ». Selon ces propositions, nous définissons une compétence comme

la capacité à mettre en oeuvre un ensemble de connaissances et de savoir-faire dans des situations variées pour atteindre les objectifs désirés.

Pour prendre en compte cette définition, nous avons spécifié dans le méta-modèle les trois objets *Competency*, *Skill* et *Knowledge*. Ces objets sont une spécialisation de l'objet *Framework_Object*, ils héritent donc des propriétés et relations définies dans la section précédente.

4.1. Les objets de type *Competency*

Un objet de type *Competency* peut avoir les relations suivantes avec des entités de nature différente :

- **hasSkill/isSkillOf** : cette relation est définie avec des entités de type *Skill* (voir plus loin). Elle exprime le fait qu'une compétence peut comprendre un ou plusieurs savoir-faire.
Exemple : la compétence "Manipuler des variables" comprend le savoir-faire "Affecter une valeur à une variable".
- **hasKnowledge/isKnowledgeOf** : cette relation est définie avec des entités de type *Knowledge* (voir plus loin). Elle exprime le fait qu'une compétence peut comprendre une ou plusieurs connaissances.
Exemple : la compétence "Manipuler des variables" comprend la connaissance "Connaître la notion de portabilité d'une variable".

4.2. Les objets de type *Knowledge*

Un objet de type *Knowledge* est défini par une notion théorique nécessaire à l'acquisition d'un savoir-faire (*Skill*). Il représente un savoir à maîtriser pour mettre en œuvre une compétence ou un savoir-faire (Roegiers, 1999).

4.3. Les objets de type *Skill*

Un objet *Skill* est défini comme une aptitude ou une capacité à faire quelque chose. Un *Skill* peut être lié à une connaissance (*Knowledge*) à travers les deux relations suivantes :

- **isRequiredBy/requires** : cette relation a la même signification que la relation du même nom définie entre objets de type *Framework_Object*. Elle exprime le fait qu'un savoir-faire nécessite une connaissance pour être maîtrisé. Cette relation est réciproque, c'est-à-dire que l'apprentissage d'une connaissance peut nécessiter la maîtrise d'un savoir-faire.
Exemple : le savoir-faire "Afficher la valeur d'une variable" requiert la connaissance "Connaître la notion de variable".
- **isUnderstoodBy/isLeverOfUnderstandingOf** : cette relation a la même signification que la relation du même nom définie entre objets de type *Framework_Object*. Elle exprime le fait que la maîtrise d'un savoir-faire facilite l'apprentissage d'une connaissance. Cette relation est réciproque, c'est-à-dire qu'une connaissance peut aider à maîtriser un savoir-faire.
Exemple : la connaissance "Connaître le concept de style" facilite la maîtrise du savoir-faire "Mettre en page un document texte".

5. Les objets de type *Resource*

Cette classe représente les ressources pédagogiques qui doivent permettre l'acquisition de compétences. Dans le projet COMPER, une telle ressource correspond à n'importe quelle ressource accessible à travers une URL.

Le modèle détaillé des ressources correspond au livrable 2.1 du lot 2.

6. L'outil d'édition de référentiel

Un outil fondé sur les technologies web a été développé pour permettre la conception de référentiels de compétences spécifiques à une discipline particulière, et conformes au méta-modèle présenté dans ce document. L'outil est accessible à cette adresse :

<https://traffic.irit.fr/comper/repository/>

L'outil a été utilisé pour concevoir les 3 référentiels correspondant à chacun des terrains du projet. Ces référentiels ont été livrés sous la forme de fichiers OWL.

Références

- Campion, M. A., Fink, A. A., Ruggeberg, B. J., Carr, L., Phillips, G. M., & Odman, R. B. (2011). Doing competencies well: Best practices in competency modeling. *Personnel psychology*, 64(1), 225-262.
- CEN. (2014). European e-Competence Framework 3.0. A common European Framework for ICT Professionals in all industry sectors (Report CWA 16234 Part 1). Available at http://ecompetences.eu/wp-content/uploads/2014/02/European-e-Competence-Framework-3.0_CEN_CWA_16234-1_2014.pdf.
- International Project Management Association. (2015). Individual Competence Baseline: For Project, Programme & Portfolio Management. International Project Management Association (IPMA).
- Kang, Y., & Ritzhaupt, A. D. (2015). A job announcement analysis of educational technology professional positions: Knowledge, skills, and abilities. *Journal of Educational Technology Systems*, 43(3), 231-256.
- Meirieu, P. (1987). *Apprendre, oui, mais comment ?* Paris, France : ESF.
- Pelap, G. F., Zucker, C. F., & Gandon, F. (2018). Semantic models in web based educational system integration. In *Proceedings of The 14th International Conference on Web Information Systems and Technologies (WEBIST)*, p. 78-89, Seville. SCITEPRESS.
- Roegiers, X. (1999). Savoirs, capacités et compétences à l'école: une quête de sens. In *Forum-pédagogies*, p. 24-31.