

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/320310837>

# Évaluation par compétences d'activités de programmation créative avec l'outil #5c21

Conference Paper · October 2017

CITATIONS

2

READS

906

2 authors:



Alexandre Lepage

Laval University

5 PUBLICATIONS 39 CITATIONS

SEE PROFILE



Margarida Romero

Université Côte d'Azur

244 PUBLICATIONS 1,638 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



LINE. Axe Créativité. [View project](#)



Intergenerational learning and creative uses of technologies [View project](#)

Lepage, A., & Romero, M. (2017). Évaluation par compétences d'activités de programmation créative avec l'outil #5c21. In *Actes du colloque CIRTA 2017* (Vol. 1). UQAM, Québec: CRIRES.

## Évaluation par compétences d'activités de programmation créative avec l'outil #5c21

Alexandre Lepage<sup>1</sup>, Margarida Romero<sup>1,2</sup>

<sup>1</sup> Université Laval, Québec, [alexandre.lepage.1@ulaval.ca](mailto:alexandre.lepage.1@ulaval.ca)

<sup>2</sup> Laboratoire d'Innovation et Numérique pour l'Éducation, Université de Nice Sophia Antipolis, France  
[margarida.romero@unice.fr](mailto:margarida.romero@unice.fr)

**Résumé:** Les activités techno-créatives engagent l'élève dans la résolution d'un problème au cours duquel il fait usage de la pensée informatique comme ensemble de stratégies qui sont basées sur les sciences du numérique. Les activités techno-créatives impliquant la programmation créative ou la robotique pédagogique présentent un potentiel pour le développement de compétences comme la résolution de problèmes et la pensée informatique. Cependant, la difficulté à évaluer ces activités techno-créatives dans une approche par compétences fait en sorte qu'elles ne sont souvent pas évaluées. Dans cet article, nous proposons un outil visant à faciliter l'évaluation de compétences par l'élève, l'enseignant ou un pair à partir d'observables définis pour chaque activité.

### Programmation créative et pensée informatique

L'apprentissage de la programmation à l'école peut se réaliser dans une grande diversité d'activités. Des tutoriels guidés aux activités techno-créatives (Romero, Davidson, Cucinelli, Ouellet, & Arthur, 2016), il existe d'importantes différences sur le niveau d'engagement des élèves dans la démarche de résolution de problèmes. La programmation créative vise à engager les participants dans une démarche critique, empathique et créative de résolution de problèmes d'une certaine complexité et authenticité, tout en faisant appel à l'usage de stratégies et de processus des sciences informatiques pour la création d'une ou plusieurs solutions.

À partir du modèle de compétence de Rey (2006) nous considérons la pensée informatique comme une compétence authentique qui s'observe en contexte. Nous avons identifié six composantes, ou compétences de deuxième degré pour la pensée informatique. Voici ces composantes :

1. **Analyse et abstraction.** Faire des choix pour interpréter la situation, par exemple en accordant de l'importance à des éléments et moins à d'autres.
2. **Organisation et modélisation.** Produire un plan de la solution à développer en identifiant les acteurs, objets, processus.
3. **Code.** Choisir un langage approprié, savoir lire un algorithme, connaître des éléments de culture numérique permettant de discuter ces concepts.
4. **Systèmes technologiques.** Il faut tenir compte des systèmes informatiques et technologiques qui vont soutenir notre programme informatique. A-t-on besoin d'un réseau? d'un serveur? d'une trousse robotique?
5. **Programmation.** Une fois le langage a été défini, l'outil et les systèmes informatiques qui vont donner une réponse aux besoins de l'étape 1 et 2, nous allons créer un programme informatique (programmation).
6. **Évaluation et amélioration itérative.** La dernière étape de la démarche correspond à un engagement dans l'évaluation du processus itératif pour améliorer le programme.

Le développement des composantes de la pensée informatique dans le cadre d'activités d'apprentissages est souvent réalisée de façon non linéaire et par des allers-retours entre chacune, dans un processus qui garde des liens étroits avec celui de la résolution collaborative de problèmes (Calder, 2010) et de l'approche de pensée *design* en éducation (Kafai, 1996).

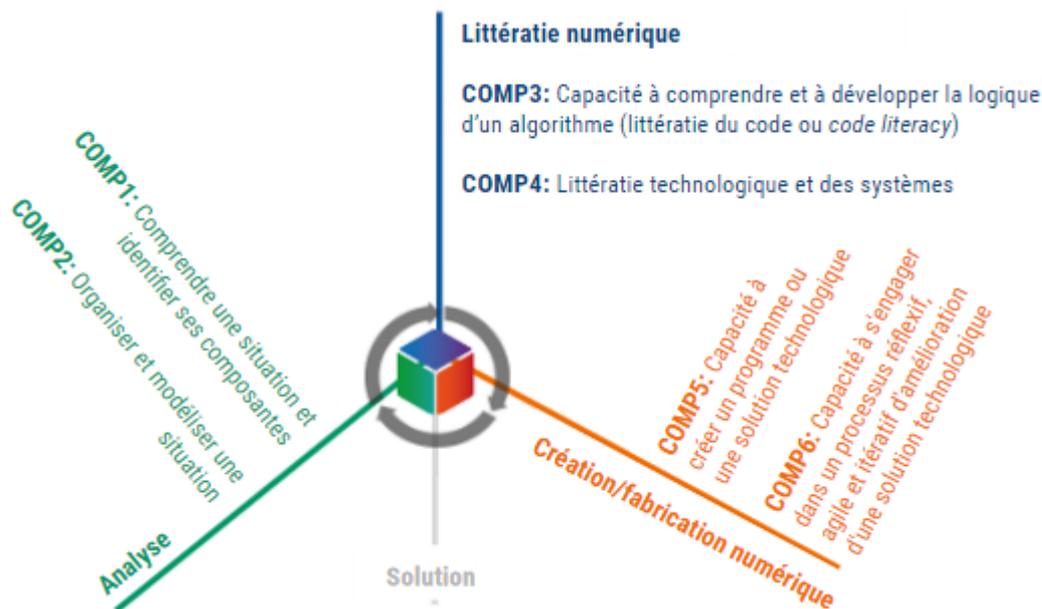


Figure 1. Six composantes de la pensée informatique.

Chacune des six composantes de la pensée informatique se déclinent en sous-composantes (compétences de premier degré). Par exemple, une des sous-composantes de C1 (Analyse et abstraction) est l'identification de la temporalité dans une situation-problème. À ce niveau, celui de la sous-composante, il doit être possible de définir concrètement à quoi cela correspond dans une tâche. Par exemple, identifier la temporalité pourrait s'observer par la production d'une ligne du temps ou d'un scénarimage adapté à une situation.

### Les défis techno-créatifs pour l'apprentissage de la programmation

Dans les activités qui font appel aux usages du numérique, le niveau d'engagement des apprenants dépend, entre autres aspects, du type d'activités qui leur sont proposées (Chi & Wylie, 2014; Romero, Laferriere, & Power, 2016). Parmi ces activités, les défis techno-créatifs sont des activités de création au cours desquelles les apprenants doivent résoudre une situation problématique d'une certaine complexité. Ils ont alors accès à différentes technologies qu'ils peuvent exploiter ou non.

Dans ce contexte, nous avons développé un ensemble de contes à coder (Romero, Roy, & Lepage, 2016). Les défis de programmation créative présentent une petite histoire mettant en scène le chat Scratch et 'Vibot le robot' (Romero & Loufane, 2016). Pour chaque défi, les apprenants doivent analyser l'histoire, identifier les éléments (personnages, décor, actions, ...) et ensuite les programmer sur Scratch en remixant le canevas qui est proposé. Ces défis créatifs doivent ensuite faire l'objet d'évaluation, ce qui nous amène à la proposition centrale : un outil permettant d'évaluer selon une approche par compétences doit tenir compte du contexte d'évaluation (Hartig, Klieme, & Leutner, 2008).

### L'outil d'évaluation de compétences #5c21

La considération de la pensée informatique comme une compétence authentique telle que définie par Rey (2006) fait qu'elle doit être observée en contexte. Ainsi, il est nécessaire de trouver une façon de transformer les sous-composantes dans des éléments observables particuliers et concrets à une situation donnée. Les énoncés de compétences, des composantes et des sous-composantes étant par essence abstraits, il serait contraire au principe de justesse de ne pas définir pour chaque situation les éléments

concrets à partir desquels peuvent être inférées les sous-composantes. Cette position nous amène à la présentation de l'outil du projet CoCreaTic #5c21 conçu spécialement pour répondre à ce problème.

L'outil #5c21 permet la création de grilles pour l'évaluation d'activités pédagogiques. Chaque grille est constituée par l'enseignant à partir de sous-composantes à évaluer dans une activité spécifique. L'enseignant choisit des sous-composantes pouvant être observées de façon concrète dans l'activité. Il doit par la suite décrire, en un énoncé concis, par quels éléments concrets peuvent être inférées les sous-composantes. Par exemple, la sous-composante intitulée « Reconnaître et adapter un algorithme existant » pourrait être décrite par cet énoncé spécifique : « Le chat glisse vers l'arbre plutôt que d'y apparaître instantanément ». Il importe ici de justifier le choix fait de situer l'évaluation au niveau des sous-composantes plutôt qu'à celui des composantes ou de la compétence entière. Il a été dit que les compétences de deuxième degré, ici appelées composantes, sont déjà des manifestations abstraites ne pouvant être inférées qu'à partir de plusieurs agencements de compétences de premier degré, ici appelées sous-composantes. L'évaluation des composantes et de la compétence entière seront donc à faire à partir d'agrégation de données recueillies d'observations des sous-composantes en contexte.

Une fois que l'enseignant a déterminé les sous-composantes à évaluer et qu'il a décrit pour chacune l'observable concret, il peut procéder à l'élaboration d'échelles pour spécifier ses attentes. Afin de rendre l'évaluation la plus objective possible, nous avons opté pour des échelles qualitatives à niveaux. L'enseignant peut déterminer un nombre de niveaux possible pour chaque observable et associer à chaque niveau un score. Reprenant l'exemple précédent, nous pourrions décliner l'observable en quatre niveaux possibles, associés à des scores allant de 1 à 4 : « Le chat apparaît instantanément à l'arbre ou ne se déplace pas », « Le chat glisse par mouvements saccadés vers l'arbre », « Le chat glisse de façon fluide vers l'arbre » et « Le chat marche progressivement vers l'arbre ».

Par la suite, la grille d'évaluation peut être utilisée par différents agents évaluateurs. Elle peut être utilisée par l'apprenant en autoévaluation, par des pairs entre eux, par un ou plusieurs enseignants ou par un chercheur.

L'outil #5c21 est en développement à deux niveaux : au niveau opérationnel et technique, c'est-à-dire que nous développons un prototype permettant d'atteindre les objectifs conceptuels fixés, et au niveau de la démarche d'évaluation par compétences. À ce jour, il a été utilisé à une dizaine de reprises dans des séminaires, formations ou activités impliquant parfois des enseignant.e.s, parfois des étudiant.e.s universitaires et parfois des élèves du primaire. Ces utilisations s'inscrivent dans une démarche itérative de bonification du prototype. Certaines données récoltées font présentement l'objet d'analyses préliminaires par lesquelles nous validerons leur pertinence et leur signification dans une perspective de recherche scientifique.

## **Discussion**

Rey (2006) propose un modèle d'évaluation des compétences en trois temps. Pour situer l'élève dans la maîtrise d'une compétence, il propose de l'exposer à la même tâche à trois reprises, chaque fois avec un niveau de guidage différent. Au premier tour, l'élève doit accomplir la tâche complexe sans aucun guidage. Il doit montrer sa capacité à agréger différentes ressources et à les orienter vers un but. Au deuxième tour, la même tâche est présentée, mais divisée en sous-tâches ciblant des composantes de deuxième degré. L'écart entre le résultat de la première tâche, non guidée, et de la deuxième tâche, découpée, permet de voir si l'élève a de la difficulté à mobiliser, de lui-même, en contexte, des compétences de premier degré (Rey, 2006). Finalement, au troisième tour, la tâche est découpée en tâches décontextualisées (par exemple, additionner des nombres sans mise en situation). Ce tour correspond à l'évaluation des compétences de premier degré. Il importe de préciser que l'outil #5c21 ne prétend pas opérationnaliser un seul modèle d'évaluation. L'évaluation des compétences de premier degré est contextualisée, même si la tâche est découpée au moment de l'évaluation. Choisir les sous-composantes à évaluer peut revenir à rejeter le choix que doit faire l'élève des ressources à mobiliser. En effet, si celles-ci sont établies d'avance, les données recueillies ne suffisent peut-être pas à inférer

toutes les caractéristiques d'une compétence authentique. De même, cette accumulation systématique de données d'autoévaluation, d'évaluation par l'enseignant et d'évaluation par les pairs mériterait d'être couplée à celle du portfolio, par exemple en laissant à l'élève le soin de choisir à quelles sous-composantes ses réalisations correspondent en se justifiant. Le rôle de l'enseignant pourrait alors être d'accepter ou de rejeter les justifications.

L'évaluation basée sur les compétences nécessite une opérationnalisation qui permette à l'enseignant d'identifier des observables concrets pour être évalués à partir de toute une série d'activités. Le développement et l'usage de l'outil #5c21 sert à collecter des données d'évaluation permettant de dégager des tendances principales et adapter l'activité professionnelle en fonction des besoins des clients.

## Références

- Calder, N. (2010). Using Scratch: An Integrated Problem-Solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, 15(4), 9–14.
- Chi, M. T. H., & Wylie, R. (2014). The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 49(4), 219–243.
- Hartig, J., Klieme, E., & Leutner, D. (2008). *Assessment of competencies in educational contexts*. Hogrefe Publishing.
- Kafai, Y. B. (1996). Learning design by making games. *Constructionism in Practice: Designing, Thinking and Learning in a Digital World*, 71–96.
- Rey, B. (2006). *Les compétences à l'école : apprentissage et évaluation* (2e éd). Bruxelles: De Boeck. Retrieved from <http://ariane.ulaval.ca/cgi-bin/recherche.cgi?qu=i2804150046>
- Romero, M., Davidson, A.-L., Cucinelli, G., Ouellet, H., & Arthur, K. (2016). Learning to code: from procedural puzzle-based games to creative programming. In *Learning and teaching innovation impacts*. Barcelona, Spain: ACUP.
- Romero, M., Laferrriere, T., & Power, T. M. (2016). The Move is On! From the Passive Multimedia Learner to the Engaged Co-creator. *eLearn*, 2016(3), 1.
- Romero, M., & Loufane. (2016). *ViBot, le robot*. Québec, QC: Publications du Québec.
- Romero, M., Roy, A., & Lepage, A. (2016). Défis de programmation créative: du conte au code avec Scratch et Vibot. CoCreaTIC. Retrieved from <https://goo.gl/bBWGIk>