

Vers une massification de l'apprentissage instrumenté de l'informatique, et une intégration des instruments et de leur évaluation

Fahima Djelil¹, Charles Boivert², Yvan Peter³, Yann Secq⁴, Julien Broisin⁵, Colin De La Higuera⁶

(1) Université de Haute-Alsace ; (2) Université Sheffield Hallam (Royaume-Uni); (3,4) Université de Lille; (5) Université Toulouse III Paul Sabatier; (6) Université de Nantes

fahima.djelil@uha.fr; C.Boisvert@shu.ac.uk; yvan.peter@univ-lille1.fr; yann.secq@univ-lille1.fr; broisin@irit.fr; cdlh@univ-nantes.fr;

1. Description et principaux objectifs

L'apprentissage instrumenté de l'informatique est un enjeu de recherche depuis de nombreuses années [1], [2], [3]. En particulier l'apprentissage de la pensée informatique et de la programmation soulève de nombreuses questions : Quels objets physiques ou numériques ? Quelles plateformes pour porter les concepts, partager le code entre les élèves et avec l'enseignant ? Comment accompagner les enseignants ? Comment évaluer ces dispositifs ?

De nombreux outils existent: chercheurs et enseignants créent des outils pour la visualisation, la définition d'algorithmes et l'initiation à la programmation [4]. Quelques initiatives de référencement de ressources existent mais ne suffisent pas à accompagner l'enseignant jusque dans la classe [5], [6]. A cela s'ajoute le manque de méthodologies et de résultats de leur évaluation [7], [8]. Or, l'évaluation de ces outils est un véritable enjeu de leur développement : comment convaincre de leur bien fondé et de leur pertinence ? Quels sont leurs bénéfices éducatifs ? Quels sont les publics qu'ils visent ? Comment accompagner les enseignants dans ce travail d'instrumentation ? Il est indispensable d'indiquer que ces outils ont été dûment évalués pour assurer leur utilisabilité et convaincre de leur utilité.

L'intégration de ces dispositifs et de leurs moyens d'évaluation faciliterait le choix d'un outil, son appropriation par l'apprenant au niveau des concepts manipulés, l'accompagnement de l'enseignant dans son utilisation, et son déploiement en classe.

2. Problèmes éducatifs concernés et bénéfiques

La pensée informatique est reconnue comme un acquis fondamental, qui ne doit pas être limité qu'aux informaticiens [9]. Elle désigne un ensemble d'habiletés utilisables

dans différentes disciplines : décomposition, identification de motifs récurrents et abstraction amenant à la définition d'un algorithme de traitement d'un problème [9]. Elle est incluse dans les savoirs fondamentaux que l'école doit transmettre aux élèves, et s'ajoute à la lecture, l'écriture et l'arithmétique dès le très jeune âge. Il s'agit de former les nouvelles générations à l'informatique afin de les préparer aux métiers de demain, et de leur donner une chance équitable pour réussir. C'est dans cette perspective qu'en France, le plan numérique pour l'éducation lancé en 2015, comprend un enseignement généralisé de l'informatique. A la rentrée 2016, ce plan touchait environ 12 millions d'élèves et 861 000 enseignants de premier et second degrés [10]. Aujourd'hui, ces enseignants sont confrontés au manque de ressources et d'accompagnement à l'utilisation de celles qui existent. De plus, les outils existants ne font pas toujours l'objet d'une évaluation, et quand elle existe elle reste dépendante du contexte de conception de ces outils. Ce qui, en pratique, rend le choix d'un outil par l'enseignant difficile.

L'intégration des outils, de méthodes et de résultats de leur évaluation offrirait un référentiel associé à des communautés de pratique, permettant le partage et le transfert des connaissances entre les enseignants, les apprenants, les chercheurs ou d'autres praticiens. Enfin, du point de vue de l'apprenant, cela permettrait d'explorer et d'évaluer des approches d'apprentissage collaboratives et ludiques, par l'utilisation de jeux, d'interfaces tangibles ou de tuteurs intelligents. Ce qui permet de favoriser le développement de connaissances et de compétences chez l'apprenant.

3. Activités

Tableau 1. Description des étapes/activités à mettre en œuvre.

Activités	Description
A1 - Curation des outils existants	Description, indexation et association d'outils avec les grains de formation qui permettront d'accompagner l'enseignant dans les concepts manipulés.
A2 - Recherche de standards d'intégration des outils	Définition et prototypage des standards permettant l'intégration d'outils complémentaires, pouvant servir, par exemple, à la collecte, la gestion et l'analyse de traces d'activités et d'interaction, à la déduction de profils d'apprenants, ou à l'adaptation des activités d'apprentissage.
A3 - Réalisation d'une cartographie d'évaluation d'outils	Cartographier les données relatives à l'évaluation d'outils, comprenant des méthodes, des méthodologies et des résultats. Cela doit servir aux chercheurs à des fins de réplique et de comparabilité d'outils afin de valider leur usage et leur qualité. Cela requiert une collaboration interdisciplinaire, mêlant des chercheurs en informatique et en sciences de l'éducation.

4. Indicateurs de succès

Tableau 2. Description des indicateurs de succès pour chaque activité prévue.

Activités	Description
A1 - Curation des outils existants	- Nombre de ressources et outils indexés. - Association effective de grains de formation associés
A2 - Recherche de standards d'intégration des outils	- Cadres pour l'architecture de services et pour la collecte des traces prenant en compte les normes existantes (e.g. xAPI [11], Caliper [12]) et les projets en cours (e.g. ANR Hubble [13])
A3 - Réalisation d'une cartographie d'évaluation d'outils	- Une taxonomie pouvant amener à une ontologie des évaluations, facilement accessible, regroupant des méthodes réutilisables et des informations sur les méthodes et les résultats d'évaluation d'outils.

Bibliographie

- [1] L. Winslow, "Programming pedagogy - a psychological overview," *ACM Sigcse Bulletin*, vol. 28, no. 3, pp. 17-22, 1996.
- [2] A. Robins, J. Rountree and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. 13, no. 2, pp. 137-172, 2003.
- [3] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting and T. Wilusz, "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," *ACM SIGCSE Bulletin*, vol. 33, no. 4, pp. 125--180, 2001.
- [4] T. Rongas, A. Kaarna and H. Kälviäinen, "Classification of tools for use in introductory programming courses," Lappeenranta University of Technology, 2004.
- [5] Pixees, "<https://pixees.fr>,".
- [6] ClassCode, "<https://pixees.fr/classcode/accueil>,".
- [7] X. López, J. Valenzuela, M. Nussbaum and C.-C. Tsai, "Some recommendations for the reporting of quantitative studies," *Computers & Education*, vol. 91, no. C, pp. 106-110, 2015.
- [8] P. Twining, R. S. Heller, M. Nussbaum and C.-C. Tsai, "Some Guidance on Conducting and Reporting Qualitative Studies," *Elsevier*, 2016.
- [9] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no.

3, p. 2006, 33-35.

- [10] Chiffres clés du système éducatif, "http://www.education.gouv.fr/cid195/les-chiffres-cles-du-systeme-educatif.html#Les_personnels".
- [11] xAPI, "<http://experienceapi.com/overview/>".
- [12] Caliper, "<https://www.imsglobal.org/activity/caliperram>".
- [13] P. Hubble, "<http://hubblelearn.imag.fr>".