Proposition d'un référentiel de formation à l'enseignement de l'informatique

Gabriel Parriaux[1] [0000−0002−8921−5459], Jean-Philippe Pellet[1,2] [0000−0001−7559−397X]
et Vassilis Komis[3] [0000−0001−6909−2765]

[1] Haute école pédagogique Vaud, Lausanne, Suisse
{gabriel.parriaux, jean-philippe.pellet}@hepl.ch
[2] École polytechnique fédérale de Lausanne, Suisse
[3] Université de Patras, Grèce
komis@upatras.gr

Résumé Nous cherchons à constituer un référentiel pour la formation des enseignant·e·s à l'enseignement de la science informatique. Sur la base d'une analyse de référentiels existants ainsi que de recherches portant sur leurs connaissances et compétences spécifiques, nous établissons une liste de sujets incontournables et en proposons une structure. Les recherches entreprises fournissent un lot important d'aspects à traiter, en particulier sur l'apprentissage de la programmation et les difficultés courantes des élèves à ce sujet. La littérature scientifique du domaine nous rappelle que l'apprentissage de la programmation constitue un domaine hautement complexe dont il s'agit de ne pas sous-estimer les difficultés.

Keywords: Informatique · Référentiel de compétences · Objectifs de formation · PCK · TPACK.

1 Introduction et contexte

Un mouvement est en cours dans diverses régions du monde pour introduire un enseignement de science informatique à l'école. Ce faisant, de nombreux curricula voient le jour qui présentent les notions que les élèves sont censés aborder, les objectifs qu'ils doivent atteindre, en fonction de leur âge.

La question de la formation des enseignant·e·s à ce nouvel enseignement est souvent présentée comme l'un des facteurs importants de réussite d'une telle introduction.

Pour autant, peu de travaux présentent de manière globale les connaissances et compétences spécifiques dont ont besoin les enseignant·e·s pour enseigner la science informatique et à notre connaissance peu de référentiels de formation pour les enseignant·e·s à la science informatique sont disponibles.

Nous proposons dans ce travail de constituer un référentiel de formation des enseignant·e·s à l'enseignement de la science informatique.

Comme point de départ, nous utilisons le canevas d'un référentiel utilisé dans une institution en charge de la formation des enseignant·e·s dont deux des auteurs sont issus, la Haute école pédagogique Vaud en Suisse, qui dispose d'une structure de base d'un référentiel en didactique de l'informatique comprenant les trois parties suivantes :

— Concevoir, mettre en œuvre et évaluer des situations d'enseignement-apprentissage en science informatique

— Disposer des connaissances de base de la science informatique

— Évaluer les enjeux technologiques et sociétaux de la science informatique

La première partie porte sur les aspects didactiques, la seconde aborde les contenus disciplinaires, alors que la troisième discute de la question spécifique des enjeux sociétaux — dont on peut se demander s'il s'agit d'un contenu disciplinaire ou d'un aspect à part entière. À noter que ce référentiel n'est pas suffisamment détaillé pour prétendre répondre aux besoins des auteurs, d'où les investigations lancées pour le compléter.

Nous nous appuyons alors sur un référentiel plus complet pour les enseignants d'informatique — le seul que nous ayons trouvé — ainsi que cinq articles de recherche portant sur des questions vives liées à l'enseignement de la science informatique. La sélection de ces articles s'est faite sur la base de l'intérêt qu'ils semblaient présenter pour les auteurs, mais ils n'ont pas la prétention de donner une vision complète de la question. Ce travail constitue donc une recherche préliminaire qu'il s'agirait d'approfondir en explorant un nombre plus important de ressources scientifiques.

2 Référentiel ISTE & CSTA pour l'enseignement de l'informatique

L'International Society for Technology in Education est une association à l'origine de plusieurs curricula pour des élèves ou pour des enseignant·e·s en lien avec le domaine du numérique. Celui qui nous intéresse est un standard de 2011 intitulé *ISTE Standards for Computer Science Educators*. Début 2019, ISTE s'est associée avec la Computer Science Teachers Association (CSTA) pour réviser ces standards. Cette révision est en cours au moment de la rédaction de cet article et un document intermédiaire est disponible en ligne, sur lequel le public est invité à formuler des commentaires (International Society for Technology in Education & Computer Science Teachers Association, 2019).

Au niveau des contenus disciplinaires, nous y retrouvons des éléments qui sont bien identifiés comme des thèmes centraux de la discipline : les systèmes et réseaux, la représentation des données, l'algorithmique et programmation. À cela s'ajoutent les éléments relatifs à la pensée informatique comme les capacités d'abstraction, de résolution de problèmes et autres. À noter que la question des impacts de l'informatique est ici considérée comme un contenu disciplinaire, alors que dans notre canevas de départ, celle-ci constitue une catégorie à part entière.

Concernant les aspects didactiques, le référentiel ISTE & CSTA nous semble rester très général et n'amène que peu de contenu.

3 *Acquisition of programming knowledge and skills* (Rogalski & Samurçay, 1990)

Le texte de Rogalski et Samurçay (1990) est centré sur l'apprentissage de la programmation et sur les difficultés rencontrées par les apprenants dans ce domaine. L'apprentissage de la programmation y est présenté comme un processus hautement complexe, faisant appel à des activités cognitives et des représentations mentales très variées.

Quatre difficultés particulières sont présentées : la question de la représentation conceptuelle de l'ordinateur ou machine notionnelle ; la compréhension des structures de contrôle (conditionnelles, itératives et récursives) ; la compréhension des notions de variables, de structures de données et de représentation des données ; l'explicitation de méthodes ou stratégies de programmation.

4 *Pedagogical content knowledge and educational cases in computer science : an exploration* (Koppelman, 2008)

L'article de Koppelman (2008) revêt un intérêt particulier, puisqu'à la différence de la plupart des articles de recherche du domaine qui portent sur des difficultés liées à l'apprentissage de la programmation, celui-ci aborde des exemples de situations liées à la représentation des données.

Les connaissances des enseignant·e·s liées à l'identification et l'analyse des problèmes que les élèves rencontrent dans leur apprentissage de l'informatique occupent la place principale. Une démarche de l'enseignant·e idéal·e est proposée : pour tout sujet, identifier les difficultés potentielles, les erreurs et les compréhensions erronées habituelles des élèves ; composer des exercices qui permettent aux élèves de se confronter à ces difficultés et de les clarifier ; anticiper les réponses correctes et incorrectes des élèves ainsi que les rétroactions à leur donner.

5 *An introduction to program comprehension for computer science educators* (Schulte, Clear, Taherkhani, Busjahn, & Paterson, 2010)

Schulte et al. (2010) présentent la *compréhension de programme informatique* comme un domaine qui doit être abordé en soi dans le cadre de l'enseignement, à part de celui de la production de programme informatique.

Ils décrivent plusieurs modèles de compréhension de programmes partageant des caractéristiques communes : ils décrivent un processus d'assimilation, des structures cognitives comme la représentation mentale du code et des connaissances de base nécessaires à la construction de ces représentations. Ces éléments peuvent tout à fait servir à construire un modèle pour l'apprentissage de la lecture et de la compréhension de programmes.

Schulte a lui-même développé l'un de ces modèles, le *Block Model*, qui pourrait constituer un outil d'analyse utile dans ce contexte.

Une liste de notions à aborder dans le cadre de l'enseignement de l'informatique est proposée : l'orientation générale d'un programme, la machine notionnelle, la notation, les structures, les capacités de débogage, la connaissance du domaine ainsi que les stratégies de lecture et de compréhension du code.

6 *Teaching programming in secondary school : a pedagogical content knowledge perspective* (Saeli, Perrenet, Jochems, & Zwaneveld, 2011)

Dans leur article, Saeli et al. (2011) font référence au modèle PCK (Pedagogical Content Knowledge) pour décrire les connaissances des enseignant·e·s d'informatique.

Comme dans l'article précédent, la nécessité est relevée de couvrir tant les connaissances relatives à la compréhension d'un programme que celles relatives à la production d'un programme. Un listing détaillé des concepts et compétences relatifs à l'informatique est proposé : les données, les instructions, la syntaxe et la sémantique, les expressions primitives, les modes de combinaison et d'abstraction. En ce qui concerne la question des difficultés courantes des élèves, une liste de celles-ci est proposée comprenant les difficultés relatives : à la machine notionnelle, à la notation, aux structures, au débogage et au changement de paradigme.

Enfin, les auteurs recommandent de sélectionner des problèmes qui soient indépendants d'un langage de programmation pour développer la pensée algorithmique.

7 *Towards a conceptualization of pedagogical content knowledge for computer science* (Hubwieser, Magenheim, Mühling, & Ruf, 2013)

Hubwieser et al. (2013) présentent une démarche qui consiste à proposer une conceptualisation des *Connaissances Pédagogiques et des Contenus* (PCK) pour l'informatique sur la base de la littérature, puis de la valider avec quelques enseignant·e·s. Il s'agit donc d'une approche normative/déductive d'abord, puis empirique/inductive ensuite. Afin de modéliser quelles sont ces *Connaissances Pédagogiques et des Contenus* (PCK) qui sont nécessaires pour enseigner l'informatique, les auteurs proposent un système bidimensionnel, une matrice qu'ils appellent le «modèle PCK» (2013, p. 3).

Sur la première dimension, cette matrice contient trois domaines d'activités pédagogiques et situations critiques anticipées : la planification et la conception de situations d'apprentissage, la réaction aux demandes des étudiants durant l'enseignement, et l'évaluation.

Sur la seconde dimension, la matrice contient quinze aspects de l'enseignement et de l'apprentissage : les contenus, le sujet, les curricula, les objectifs des leçons, les activités extracurriculaires, la science, les méthodes d'enseignement, les concepts à enseigner qui sont spécifiques au sujet, les éléments à enseigner spécifiques, les médias et le matériel d'enseignement, l'hétérogénéité dans le contexte du sujet traité, la cognition des élèves, la perspective de l'enseignant·e, le développement de l'école et le système éducatif.

Il est intéressant de noter que les trois domaines de la première dimension correspondent globalement aux trois phases de l'activité de l'enseignant de Durand, avec la réserve que le second domaine de « la réaction aux demandes des étudiants » chez Hubwieser et al. semble plus restreint que la phase d'animation chez Durand, qui porte sur la totalité de ce qui se passe en classe durant une leçon et pas seulement sur les rétroactions de l'enseignant.

Il est plus difficile de se prononcer sur la seconde dimension du modèle et les quinze aspects de l'enseignement et de l'apprentissage dont on peut avoir parfois du mal à comprendre la signification et la pertinence.

8 Référentiel de formation

Sur la base de l'analyse des ressources et des réflexions qui précèdent, un référentiel de formation est proposé qui constitue l'objet du poster cité en titre et qui sera présenté lors du colloque.

Références

Durand, M. (1996). *L'enseignement en milieu scolaire*. Presses universitaires de France.

Hubwieser, P., Magenheim, J., Mühling, A., & Ruf, A. (2013). Towards a conceptualization of pedagogical content knowledge for computer science. In *Proceedings of the ninth annual international acm conference on international computing education research* (pp. 1–8).

International Society for Technology in Education. (2011). *ISTE Standards for Computer Science Educators.* [Page Web]. Accès : https ://www.iste.org/standards/for-computer-science-educators (page consultée le 24 octobre 2019).

International Society for Technology in Education, & Computer Science Teachers Association. (2019). *CSTA / ISTE Standards for Computer Science Educators — Second Draft for Public Feedback.* [Page Web]. Accès : http ://bit.ly/CSEducatorStandards-Draft2 (page consultée le 24 octobre 2019).

Koppelman, H. (2008). Pedagogical content knowledge and educational cases in computer science : An exploration. In *Proceeding of the informing science and it education conference.*

Rogalski, J., & Samurçay, R. (1990). Acquisition of programming knowledge and skills. In *Psychology of programming* (pp. 157–174). Elsevier.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school : A pedagogical content knowledge perspective. *Informatics in Education*, *10*(1), 73–88.

Schulte, C., Clear, T., Taherkhani, A., Busjahn, T., & Paterson, J. H. (2010). An introduction to program comprehension for computer science educators. In *Proceedings of the 2010 iticse working group reports* (pp. 65–86).