

Sticef

*Sciences et technologies de l'information et de la communication
pour l'éducation et la formation*

Volume 26, 2019



Sticef

Sticef

Volume 26
2019

© ATIEF, 2019

ISBN 978-2-901384-02-1

DOI:10.23709/sticef.26.1 en ligne sur www.sticef.org

Le Code de la propriété intellectuelle n'autorisant, aux termes des paragraphes 2 et 3 de l'article L. 122-5, d'une part, que les « *copies et reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective* » et, d'autre part, sous réserve de mention du nom de l'auteur et de la source, que « *les analyses et les courtes citations justifiées par le caractère critique, polémique, pédagogique, scientifique ou d'information* », « *toute représentation ou reproduction totale ou partielle faite sans le consentement de l'auteur, ou de ses ayants droit ou ayants cause, est illicite* » (article L. 122-4). Une telle représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.



Sommaire

Sébastien GEORGE • Éditorial 7

Article de recherche

**Joris FELDER • Méthode d'analyse et de modélisation
des environnements personnels d'apprentissage..... 9**

Rubrique

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD • Pensée
informatique : points de vue contrastés39**

Comités63



Éditorial

Volume 26

► **Sébastien GEORGE** (Rédacteur en Chef de STICEF)

La formation en ligne s'est retrouvée du jour au lendemain propulsée sur le devant de la scène par la situation sanitaire COVID-19, notamment pendant la période de confinement. Beaucoup ont pu se rendre compte que l'enseignement à distance n'avait rien d'évident et que, bien que les environnements informatiques soient certes essentiels, les enseignants demeuraient au cœur des dispositifs pédagogiques. Cette situation apporte certainement une autre dimension aux défis que doivent relever les recherches en EIAH. Les enjeux se situent tant au niveau de la conception des outils supports aux différents acteurs qu'au niveau de l'analyse des pratiques et usages. Plus que jamais, nous avons besoin de partager les travaux scientifiques pour faire avancer les connaissances de notre domaine. La revue STICEF apporte sa contribution en publiant en accès libre des articles rendant compte de recherche sur le thème du numérique pour l'éducation et la formation, avec la volonté de croiser les regards disciplinaires.

L'année 2019 aura vu peu d'articles publiés dans la revue STICEF. Ce faible nombre est conjoncturel car, d'une part, il n'y a pas eu de publication de numéro spécial et, d'autre part, un certain nombre d'articles n'a malheureusement pas passé l'étape de validation par les relecteurs et les membres du comité de rédaction. Que les auteurs se rassurent cependant, les statistiques montrent que STICEF se situe dans les taux d'acceptation standards d'une revue sélective. Sur la période de 3 ans entre 2017 et 2019, le taux de sélection est d'environ 23% pour la catégorie « articles de recherche ». En ne prenant pas en compte dans le calcul les articles qui sont directement rejetés (car ne répondant pas aux attentes ou hors thématique) nous arrivons à un taux d'acceptation de 40%. Pour la catégorie « rubriques », le taux d'acceptation est d'environ 50% sur la même période.

Contenu du volume 26

L'article de J. Felder décrit un cadre méthodologique pour analyser et exploiter des Environnements Personnels d'Apprentissage (EPA). En suivant une démarche recherche-développement, un modèle générique d'EPA est proposé puis l'auteur décrit son application avec 15 étudiants inscrits en troisième année d'un programme de Bachelor en psychologie. Les résultats montrent une bonne expressivité des modèles conçus ainsi qu'une certaine simplicité d'interprétation. L'approche est globalement prometteuse comme outil méthodologique pour des chercheurs et des ingénieurs pédagogiques.

La rubrique proposée par B. Drot-Delange, J.-P. Pellet, Y. Delmas-Rigoutsos et É. Bruillard concerne des analyses de la notion de pensée informatique. Ces analyses font suite à une table ronde qui s'est déroulée lors du colloque Didapro 7 - DidaSTIC en 2018. Les auteurs adoptent 3 points de vue : historique, informatique et épistémologique. Ces regards croisés permettent de contribuer à mieux cerner ce qu'est la pensée informatique et comment elle peut être enseignée.

Nous souhaitons dédier ce volume à la mémoire de Jacques Perriault qui nous a quittés en septembre 2019. Il a beaucoup compté pour notre communauté, ses nombreux écrits précurseurs concernant l'analyse des usages des innovations techniques de communication ont inspiré plusieurs générations de chercheurs.



Méthode d'analyse et de modélisation des environnements personnels d'apprentissage

► **Joris FELDER** (Université de Fribourg)

■ **RÉSUMÉ** • Cette contribution méthodologique fournit au chercheur ou à l'ingénieur pédagogique désireux d'étudier ou d'exploiter les environnements personnels d'apprentissage (EPA) une méthode pour les analyser et les modéliser. Par une démarche de recherche-développement, un modèle générique et un langage de modélisation des EPA ont été conçus en visant les critères de qualité suivants : expressivité des instances d'EPA figurées par la modélisation, simplicité d'interprétation par les apprenants, portée ontologique, pragmatisme d'usage comme méthode de recherche et comme outil d'ingénierie pédagogique. La méthode a été développée dans le cadre d'une recherche longitudinale auprès de 15 étudiants universitaires, pour laquelle nous avons modélisé 60 instances d'EPA. Nous l'illustrons par un cas d'application et nous discutons sa qualité, ses limites et ses perspectives.

■ **MOTS-CLÉS** • Environnement personnel d'apprentissage, modélisation, méthode d'analyse

■ **ABSTRACT** • *This methodological contribution provides the researcher or educational engineer interested in studying or exploiting personal learning environments (PLE) with a method for analyzing and modeling them. Through a research-development approach, a generic model and an PLE modeling language were designed with the following quality criteria in mind: Expressiveness of PLE instances represented by the modeling, simplicity of interpretation by learners, ontological significance, pragmatism of use as a research method and as an educational engineering tool. The method was developed through a longitudinal study with 15 university students for which we modeled 60 instances of PLEs. We illustrate it with an application case and discuss its quality, limitations and perspectives.*

■ **KEYWORDS** • *Personal learning environment, modelling, analysis method*

1. Introduction

Cette contribution a pour but de fournir des concepts, une méthode et des instruments pour analyser et modéliser les environnements personnels d'apprentissage (EPA). Elle s'adresse au chercheur, à l'ingénieur pédagogique, à l'enseignant et à l'apprenant désireux d'exploiter ou d'étudier le concept d'EPA. S'il existe différentes conceptualisations de l'EPA (Henri, 2014), nous l'abordons ici comme une réalité subjective au sens donné par Väljataga et Laanpere (2010), c'est-à-dire comme la représentation que se fait l'individu de l'ensemble de ses instruments et de son projet d'apprentissage. Cette conceptualisation est vue comme une opportunité de renouveler la compréhension de l'apprenant d'aujourd'hui (Charlier, 2017 ; Felder, 2017 ; Henri, 2014 ; Heutte, 2014 ; Roland et Talbot, 2014) dans une « écologie d'usages » des instruments (Poizat et Durand, 2017, p. 37). Afin de saisir ce phénomène, des recherches sont menées pour comprendre sa nature, son émergence, ses tensions, ou encore ses effets sur l'individu, le dispositif d'apprentissage et la production et la transmission des savoirs. D'autres démarches de conception, d'enseignement ou de recherche consistent à exploiter le concept d'EPA comme une stratégie pédagogique – voire comme un dispositif – dans des situations d'enseignement-apprentissage visant à rendre l'apprenant plus autonome. Que l'on recoure à l'EPA dans une approche de recherche, dans des démarches pédagogiques ou de conception, les acteurs de la formation que sont les chercheurs, les développeurs, les enseignants et les apprenants ont besoin d'une instrumentation, c'est-à-dire de « concepts, méthodes et instruments d'analyse de l'activité propre et de celle d'autrui. » (Poizat et Durand, 2017, p. 36). Il s'agit d'être en mesure de capter et de rendre perceptible cet EPA qui diffère d'un individu à l'autre, qui évolue constamment et qui n'est, pour l'essentiel, pas visible.

Or, il s'avère que la recherche scientifique peine encore à concevoir, implémenter et évaluer une telle instrumentation. Une approche explorée par la recherche¹, consiste à modéliser les EPA et à les représenter visuellement, par l'apprenant lui-même (Felder, 2017 ; Mailles-Viard Metz *et al.*, 2017 ; Wilson *et al.*, 2015) ou par le chercheur (Roland et Talbot, 2014 ; Martinlade et Dowdy, 2016 ; Schaffert et Kalz, 2010). Ces approches n'utilisent aucun formalisme ou langage de modélisation formalisé. Les

1 Mais aussi par des enseignants et des apprenants : pour un aperçu, rechercher « EPA » ou « PLE » dans un moteur de recherche.

modèles d'instance d'EPA ainsi obtenus recourent à des modes de représentations variés – textes, dessins, nombres ou flèches – dont l'expressivité sémantique et conceptuelle (Maisonasse *et al.*, 2010) n'est généralement pas explicitée. En conséquence, ces modélisations ne sont comparables et analysables que de manière très limitée. Le sens des éléments présents dans les modèles est sujet à une interprétation hasardeuse et peu intelligible pour les acteurs non-experts. Finalement, les méthodes de recherche ne sont pas reproductibles, ce qui pose des problèmes de validité scientifique, mais aussi de capitalisation de la recherche.

Fort de ces constats, nous avons mené une recherche-développement visant à concevoir une méthode d'analyse et de modélisation des EPA. Pour ces développements, nous avons fixé les quatre critères de qualité suivants: 1) pragmatisme d'usage comme méthode de recherche et comme outil d'ingénierie pédagogique, 2) expressivité des instances d'EPA figurées par la modélisation, 3) simplicité d'interprétation par les apprenants, 4) portée ontologique des modèles. Au terme de nos développements, la méthode de modélisation des environnements personnels d'apprentissage (MEPA) se réalise en cinq étapes qui exploitent ses deux composantes: un modèle générique de l'EPA et un langage de modélisation des EPA. La première composante formalise les concepts et les liens sémantiques utiles à l'analyse et à l'interprétation des EPA. La deuxième est un système de symboles permettant de modéliser les données avec un haut niveau d'expressivité, en fournissant plusieurs points de vue sur les données.

Dans la suite de cet article, nous exposons la démarche de recherche-développement que nous avons suivie. Puis nous présentons la méthode MEPA et ses deux composantes. Nous illustrons son application à partir de données collectées auprès d'un étudiant universitaire. Avant de conclure, nous discutons des limites et des perspectives de la méthode présentée.

2. Démarche de recherche-développement

Dans cette section, nous décrivons tout d'abord la démarche de recherche-développement suivie pour concevoir la méthode MEPA et ses deux composantes. Notre démarche exploite également des outils et des langages qui sont présentés en section 2.2 *Langages et outils*, car ils sont porteurs de sens et contribuent aux résultats de notre recherche.

étudiants, correspondant à des situations ou des moments différents (N=60).

En parallèle, nous avons développé la méthode MEPA et ses deux composantes. La grille d'analyse catégorielle (L'Écuyer, 1990) découle du modèle générique de l'EPA. Elle a été élaborée selon une approche mixte : certaines catégories sont prédéfinies, d'autres émergent des données. Cette approche nous a amené à retourner à la littérature en cours d'analyse pour développer le modèle générique de l'EPA - et donc par récursivité, la grille d'analyse catégorielle. Dans le même temps, nous avons développé le langage de modélisation des EPA, de manière itérative. Plusieurs essais ont été prototypés avant d'aboutir. Au terme de nos développements, nous avons révisé nos analyses de données et nos modélisations pour aboutir aux 60 modèles d'instances d'EPA.

Enfin, nous avons mené un cinquième entretien afin de procéder à une validation intersubjective d'une partie des modèles d'instances d'EPA. Nous avons choisi de retenir le modèle de chaque étudiant après 12 semaines de cours (N = 15), car ceux-ci étaient les plus riches et complexes. L'entretien s'est déroulé trois mois après l'examen du cours et a permis d'évaluer la portée pragmatique de notre solution. Lors de cet entretien, nous avons présenté à chaque étudiant son modèle d'EPA après 12 semaines de cours. Cette validation intersubjective nous a amené à apporter certains ajustements à notre méthode de modélisation et au langage de modélisation.

2.2. Langages et outils mobilisés

Dans cette section, nous précisons les langages de modélisation et les outils que nous avons retenus pour les développements, car ils sont porteurs de connaissances et concourent aux résultats présentés.

Le langage utilisé pour créer le modèle générique de l'EPA est l'Ontology Web Language (OWL), « un langage du web sémantique développé pour représenter des connaissances riches et complexes à propos de choses, de groupes de choses et de relations entre les choses (...) sous forme d'ontologies » (W3C, 2012). Ce langage permet un raisonnement selon l'hypothèse du monde ouvert : ce n'est pas parce qu'une information n'est pas connue que cette information est fausse. Cette hypothèse est pertinente pour la modélisation de l'EPA lorsqu'il est conceptualisé comme un phénomène subjectif. OWL permet également qu'une même donnée empirique puisse être associée à différents concepts.

De plus, le recours aux ontologies présente une véritable opportunité pour la capitalisation de la recherche sur les EPA. Celle-ci étant pluridisciplinaire et les cadres conceptuels mobilisés multiples, elle doit se doter d'un moyen pour expliciter les relations entre ces concepts. Avec OWL, il est possible de formaliser en quoi des concepts se disjointent, se complètent, s'unissent ou encore se rejoignent (intersections).

Destiné principalement à être utilisé par des agents informatiques, le langage OWL est textuel et ne prévoit pas d'éléments graphiques pour représenter visuellement le modèle. Afin de faciliter le développement, la communication et l'interprétation de notre modèle générique, nous avons eu recours à l'éditeur G-MOT. Cet éditeur graphique mobilise le langage de modélisation par objet typé (MOT) pour représenter visuellement des ontologies OWL et pour générer une version textuelle exploitable par les agents informatiques (Paquette, 2002). Cette solution permet de formaliser et de représenter les informations syntaxiques et sémantiques conformément à notre conception de l'EPA.

3. Méthode de modélisation des EPA

La méthode MEPA est composée de cinq procédures que l'on pourra suivre de façon linéaire dans le cas d'une recherche ou que l'on pourra opérer en parallèle dans le cas d'une démarche de supervision par exemple.

- **1) Collecte de données.** Cette procédure peut avoir recours à l'entretien, au questionnaire, à l'observation ou à toute autre méthode de récolte de données.

- **2) Analyse de données.** Appliquer la grille d'analyse construite à partir du modèle générique de l'EPA pour catégoriser les données. Il est possible de recourir à une grille simplifiée en utilisant uniquement les concepts de haut niveau. Toutefois, les instances d'EPA générées seront moins expressives.

- **3) Reformulation.** Lorsque c'est nécessaire, reformuler les données pour les rendre intelligibles et intégrables aux éléments visuels lors de la procédure de modélisation suivante. Par exemple, les données collectées par entretien correspondant à un schéma d'apprentissage peuvent être volumineuses et éparées : il faudra en générer une reformulation condensée.

- **4) Modélisation de l'EPA.** Transposer les données catégorisées et les reformulations vers les éléments graphiques du langage de modélisation. Le processus peut être automatisé, par exemple en combinant les

outils Maxqda pour l'analyse catégorielle (www.maxqda.com), Neo4 pour la gestion des données (www.neo4j.com), yEd pour la production des graphes (www.yworks.com/yed-live/) recourant aux fichiers SVG du langage de modélisation de l'EPA. La démarche peut être faite manuellement à l'aide d'un outil de dessin SVG.

- **5) Validation.** Valider l'EPA modélisé, soit par validation intersubjective avec chacun des sujets, soit par validation inter-chercheurs, soit par le sujet lui-même qui aurait été instruit aux différents concepts du modèle générique de l'EPA.

En préliminaire, on procédera à la description de la situation à laquelle se rapporte la modélisation de l'EPA. Cette description recourt aux concepts du niveau 0 du modèle générique de l'EPA.

4. Modèle générique de l'EPA : première composante de la méthode MEPA

Le modèle générique de l'EPA fournit le cadre conceptuel, sémantique et syntaxique nécessaire à l'analyse (étape 2) et à la modélisation (étape 4) des données collectées. Il sert également à interpréter les modèles d'instances produits. Nous le présentons en suivant son organisation en cinq sous-modèles répartis en trois niveaux.

4.1. Niveau 0 : articulation de l'activité observée, de l'EPA et de sa modélisation

Le niveau 0 du modèle générique de l'EPA est composé d'un modèle (cf. figure 2) qui articule conceptuellement l'activité d'apprentissage observée, l'EPA et sa modélisation.

Système d'activité d'apprentissage. La modélisation d'un EPA s'inscrit dans l'observation d'un système d'activité d'apprentissage au sens d'Engeström (1999). Cette conception systémique de l'activité fournit un cadre pour étudier l'EPA en situant l'apprenant dans un système d'activité d'apprentissage constitué du sujet, de l'objet, des artefacts médiateurs, des règles, de la communauté et de la division de la tâche (Felder, 2017). Dans ce système, le sujet est un apprenant qui construit et régule son EPA, situant l'EPA comme produit de l'activité, au même titre que les apprentissages (*ibidem*).

Projet personnel d'apprentissage. L'objet du système d'activité d'apprentissage observé est la représentation du projet personnel d'apprentissage par l'apprenant, au sens de Våljataga et Laanpere (2010). Dans cette activité, l'apprenant réalise un projet personnel d'appren-

tissage. Il s'agit d'un projet d'action à conduire dont l'acteur fait un usage opératoire, c'est-à-dire qu'il « cherche à développer la maîtrise au moins partielle d'une action gouvernée par un souci d'efficacité » (Boutinet, 2002, p. 223). Ce projet n'est pas forcément conscientisé ni intentionnel pour autant, mais peut l'être ou le devenir.

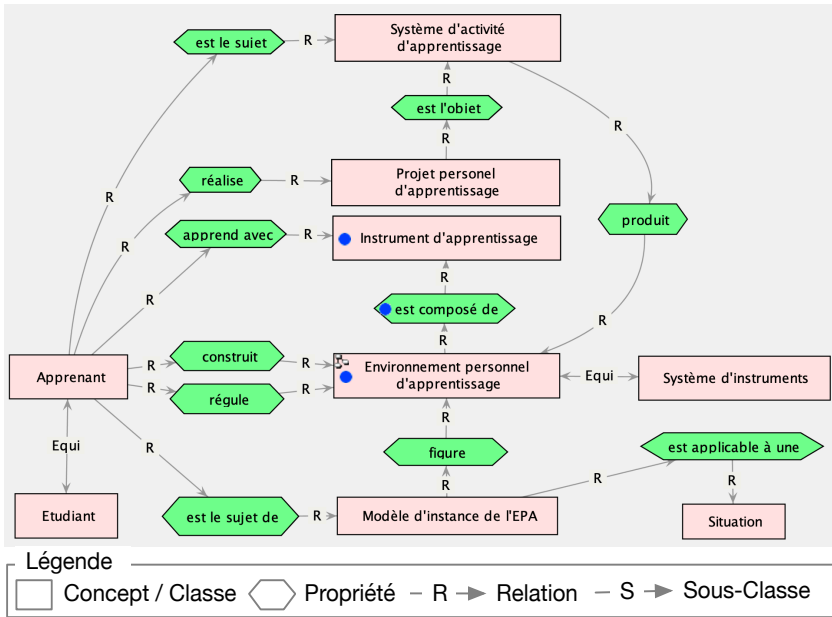


Figure 2 • Modèle générique de l'EPA : niveau 0

Instrument d'apprentissage². Dans ce système d'activité d'apprentissage, l'apprenant apprend avec des instruments d'apprentissage. À l'instar de nombreux auteurs, nous retenons la conception de l'instrument d'apprentissage au sens de l'approche instrumentale de (Charlier, 2014 ; Felder, 2017 ; Fluckiger, 2014 ; Rabardel, 1995 ; Roland et Talbot, 2014). Cette approche anthropocentrée permet « une réflexion théorique et un examen empirique des relations hommes-systèmes techniques centrées sur l'homme, vues du point de vue de celui-ci lorsqu'il est engagé dans des activités et des actions réelles, situées dans leurs contextes au travail, en formation ou dans la vie quotidienne »

² Lorsqu'un même élément se trouve sur plusieurs sous-modèles, le logiciel MOT marque d'une pastille bleue l'élément de référence et d'une pastille rouge l'élément référencé.

(Rabardel, 1995, p. 31). Il s'agit d'une analyse basée sur la représentation que se font les sujets eux-mêmes de leur environnement, approche cohérente avec la conceptualisation de l'EPA exprimée ci-dessous.

Environnement personnel d'apprentissage. Ce modèle générique de l'EPA s'appuie sur le postulat de Våljatage et Laanpere (2010) que nous avons repris (Felder, 2017) et qui considère que l'EPA est la représentation que se fait un individu de ses instruments d'apprentissage et de son projet d'apprentissage. Selon cette conceptualisation, l'EPA est une réalité subjective (Henri, 2014).

Modèle d'instance de l'EPA. Dans l'observation de ce système d'activité d'apprentissage dont l'apprenant est le sujet, la modélisation se focalise sur l'environnement personnel d'apprentissage. Chaque modèle d'instance de l'EPA est applicable à une situation d'apprentissage. Ainsi, la modélisation de l'EPA de l'étudiant est à comprendre en relation avec la représentation que l'étudiant se fait de son projet personnel d'apprentissage, qui sera de fait plus ou moins distinct du projet d'apprentissage conçu par les concepteurs du dispositif de formation.

4.2. Niveau 1 : l'EPA composé d'instruments constitués d'un schème et d'artefacts

Le niveau 1 du modèle générique de l'EPA est composé d'un sous-modèle (cf. figure 3) qui étend la définition de l'EPA comme un système d'instruments d'apprentissage exposée au niveau 1, en précisant qu'un instrument est constitué d'un schème et de quatre types d'artefacts.

Un EPA composé d'instruments d'apprentissages. Selon Rabardel, « un même schème d'utilisation peut s'appliquer à une multiplicité d'artefacts (...) inversement, un artefact est susceptible de s'insérer dans une multiplicité de schèmes d'utilisation qui vont lui attribuer des significations et des fonctions différentes. » (Rabardel, 1995, p. 4). L'artefact renvoie, dans une acception très large, aux produits transformés par l'activité humaine, qu'ils soient matériels, immatériels (numériques) ou symboliques. Dans le champ des EPA, Nieto et Dondarza se réfèrent à l'expression d'artefact digital pour des artefacts aussi divers que « des vidéos, des infographies, des présentations, des activités interactives et des tâches de tous types », « des applications, des outils et des services » (Nieto et Dondarza, 2016, p. 51). Dans de nombreux articles (Aladjem et Nachmias, 2013; Caron *et al.*, 2014; Denis et Joris, 2013; Hoechsmann et DeWaard, 2015), le terme d'artefact renvoie dans ces

textes autant à l'artefact utilisé (l'outil de traitement de texte, l'outil de création de présentation, etc.) qu'à l'artefact produit (le texte, la présentation, l'image, etc.) par l'apprenant dans son activité d'apprentissage.

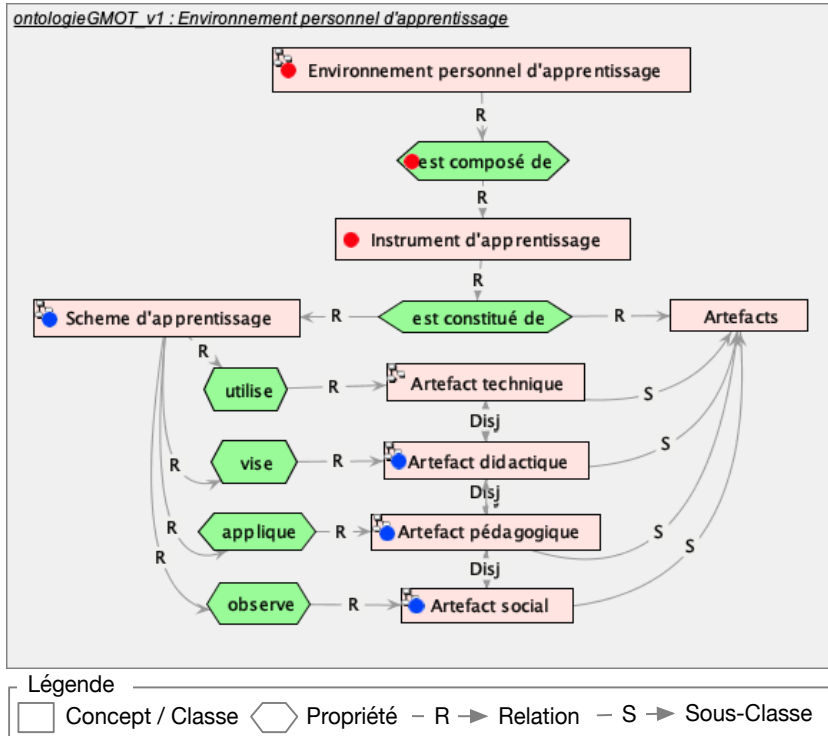


Figure 3 • Modèle générique de l'EPA : système d'instruments

Quatre sortes d'artefacts. Selon Marquet (2005), l'activité d'apprentissage instrumentée entremêle trois sortes d'artefacts (lien S sur la figure 3) - artefact technique, artefact didactique, artefact pédagogique - auxquels nous avons ajouté l'artefact social (Felder, 2014, 2017). Les définitions de ces concepts sont détaillées plus loin, pour nous concentrer ici sur leur articulation sémantique. La reconnaissance de cet entremêlement doit permettre de dépasser la seule observation des outils non intégrés et de percevoir l'EPA comme un système d'instruments. Au niveau sémantique, ce modèle propose que, dans son instrumentation et au travers d'un schéma d'apprentissage, l'étudiant *utilise* un artefact technique, *vise* l'apprentissage d'un artefact didactique, *applique* un artefact pédagogique pour apprendre et *observe* un artefact social.

Schème d'apprentissage. Eu égard à la distinction de quatre sortes d'artefacts, l'expression *schème d'utilisation* ne paraît pas cohérente. C'est pourquoi l'expression *schème d'apprentissage* est proposée pour désigner « le canevas général qui peut se reproduire en des circonstances différentes et donner lieu à des réalisations variées » (Rabardel, 1995, p. 74). Cette distinction exprime la complexité de l'apprentissage au-delà de la seule utilisation de l'artefact technique. Elle permet de considérer par exemple les artefacts didactiques en l'exprimant avec un syntagme spécifique : [schème d'apprentissage]→[vise]→[artefact didactique].

4.3. Niveau 2 : modèles des instruments d'apprentissage

Le niveau 2 du modèle générique de l'EPA est composé de trois sous-modèles : l'artefact didactique (4.3.1), l'artefact pédagogique (4.3.2) et l'artefact social (4.3.3).

4.3.1. Modèle de l'artefact didactique

L'expression *artefact didactique* est utilisée pour désigner « les objets disciplinaires enseignés » (Marquet et Leroy, 2004, p. 2) et « les connaissances structurées » (Vázquez-Cano, 2016, p. 67-68). Ainsi, ce sous-modèle (cf. figure 4) précise l'artefact didactique par deux sous-concepts : celui de connaissance et celui de compétence cible.

L'articulation des connaissances et des compétences ainsi que leur déclinaison en une typologie de compétences et de connaissances sont issues de la Méthode d'Ingénierie des Systèmes d'Apprentissage (MISA) (Paquette, 2002). Selon cet auteur, une compétence est la capacité à mobiliser une habileté par rapport à une connaissance. L'habileté peut s'appliquer à une ou plusieurs connaissances. La typologie recourt à dix « sortes » d'habiletés principales (liens S sur la figure 4) dont certaines sont déclinées en habiletés plus précises. Les connaissances sont de quatre sortes. Ces habiletés et compétences sont listées dans la figure 4.

Enfin, ce sous-modèle comporte un lien avec le sous-modèle de l'artefact pédagogique par l'intermédiaire du concept « forme de représentation des informations ». Ce dernier n'est pas un artefact didactique (absence de lien « S »). Sa présence dans ce sous-modèle exprime qu'une connaissance (artefact didactique) visée est *représentée sous une forme de représentation des informations* (artefact pédagogique) ou son équivalent, une *forme de médiatisation des connaissances*.

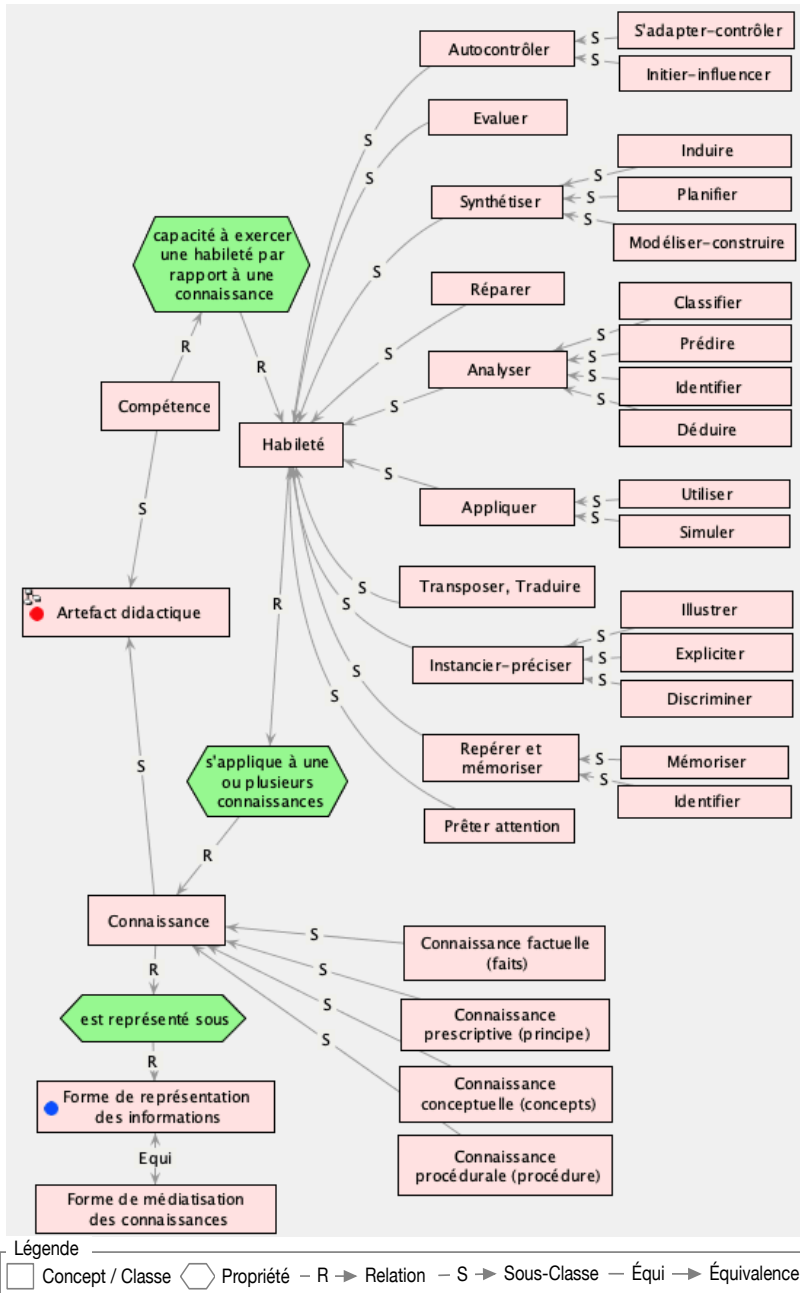


Figure 4 • Modèle générique de l'EPA : artefact didactique

4.3.2. Modèle de l'artefact pédagogique

Le sous-modèle de l'artefact pédagogique (cf. figure 5) recourt à la typologie de stratégies cognitives et métacognitives de Bégin (2008). Au moment de sa revue de la littérature, il existe selon cet auteur six sortes de stratégies cognitives de traitement de l'information : sélectionner, repérer, décomposer, comparer, élaborer, organiser. Les stratégies cognitives d'exécution sont au nombre de quatre : évaluer, vérifier, produire, traduire (vulgariser). Les stratégies métacognitives sont, selon Bégin, de deux sortes : anticiper et s'auto-réguler.

Dans ce sous-modèle, nous utilisons encore l'expression « forme de représentation des informations » pour désigner « les objets médiateurs du savoir (...) comme le langage, les formalismes et les techniques visuo-figuratives, les mnémotechniques » (Marquet et Leroy, 2004, p. 3) ou des ressources (Trouche, 2005). Cette sorte d'artefact pédagogique est relié avec le modèle de l'artefact didactique. Par exemple, il correspond à des éléments tels qu'un article scientifique, un tableau, une animation, un enregistrement audio, une note personnelle, ou encore un résumé.

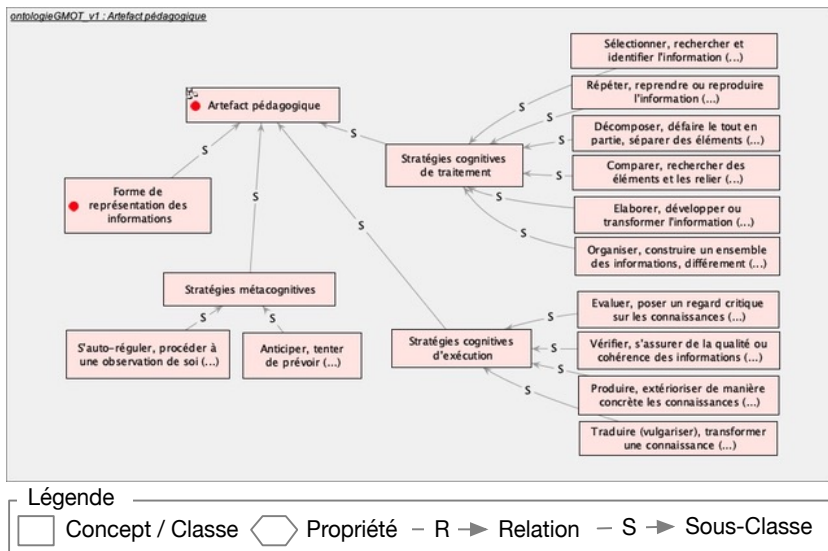


Figure 5 • Modèle générique de l'EPA : artefact pédagogique

4.3.3. Modèle de l'artefact social

Le concept d'artefact social est vu dans le champ des sciences des systèmes d'information comme l'ensemble des interactions ou des relations entre les individus et les objets sociaux persistants tels que les institutions, les rôles, les lois ou les interactions uniques telles que les décisions (Vartiainen et Tuunanen, 2016, p. 1268) (notre traduction). Alter (2015) rappelle, à la suite de Guarino, Bottazzi, Ferrario et Sartor, que « dans les systèmes sociotechniques complexes, les artefacts sociaux sont des normes et des institutions déterminant ce qui doit être fait et gouvernant les obligations, les buts, les pouvoirs » (Guarino *et al.*, 2012, p.2). L'auteur exprime la difficulté de distinguer l'artefact social de l'artefact technique, car tout artefact créé par l'homme contient obligatoirement une dimension sociale. Cependant, la conceptualisation de l'instrument d'apprentissage exprime cette imbrication du social avec le technique ou encore du pédagogique avec le didactique (cf. figure 6).

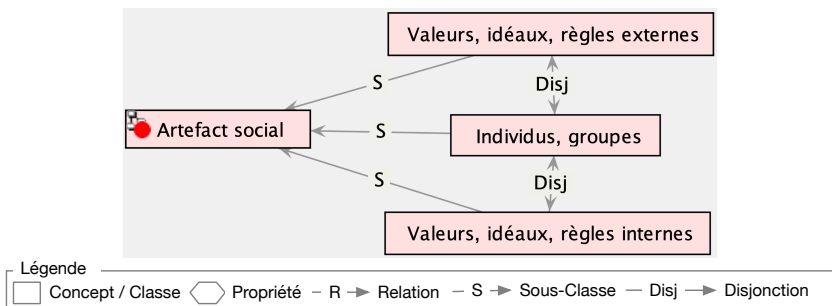


Figure 6 • Modèle générique de l'EPA : artefact social

5. Langage de modélisation des EPA, deuxième composante de la méthode MEPA

Dans cette partie, nous présentons le langage de modélisation des EPA. Il s'agit d'un système de formalismes et de symboles graphiques (cf. figure 7) qui est en relation avec une sélection de concepts du modèle générique de l'EPA. Ce langage résulte d'une hybridation du langage MOT, du modèle générique de l'EPA et d'attributs graphiques que nous avons conçus. Au total, ce langage de modélisation des EPA est composé de 42 éléments et 4 sortes de liens (cf. figure 8).

Symboles relatifs au niveau 1 du modèle générique. Le langage de modélisation des EPA a pour but de représenter l'EPA comme un système d'instruments, tel que formalisé au niveau 1 du modèle générique de l'EPA. Le système de symboles (cf. figure 7) recourt à la forme ovale pour figurer un schème et à la forme rectangulaire pour figurer un artefact. Quatre couleurs distinguent ensuite les sortes d'artefacts : le vert correspond à l'artefact didactique ; l'orange à l'artefact pédagogique ; le bleu à l'artefact technique ; et le violet à l'artefact social. La couleur jaune est associée au schème d'apprentissage en redondance à la forme ovale. Aussi, quatre flèches servent à relier le schème d'apprentissage aux quatre types d'artefacts, afin d'exprimer le concept d'instrument d'apprentissage. Les labels des flèches renvoient aux liens sémantiques entre concepts tels qu'exprimés dans le modèle générique (*vis* un artefact didactique, *utilise* un artefact technique, *applique* un artefact pédagogique, *observe* un artefact social).

Symboles relatifs au niveau 2 du modèle générique. En relation aux sous-modèles de niveau 1 du modèle générique de l'EPA, huit symboles distinguent les classes principales des quatre types d'artefacts. Un label en haut de la forme renvoie aux différents items de haut niveau des typologies décrites dans le modèle générique de l'EPA. Un label au bas de la forme renvoie aux items de bas niveau de ces typologies. Au centre, est réservé un emplacement pour inscrire la représentation de l'étudiant dans une forme succincte.

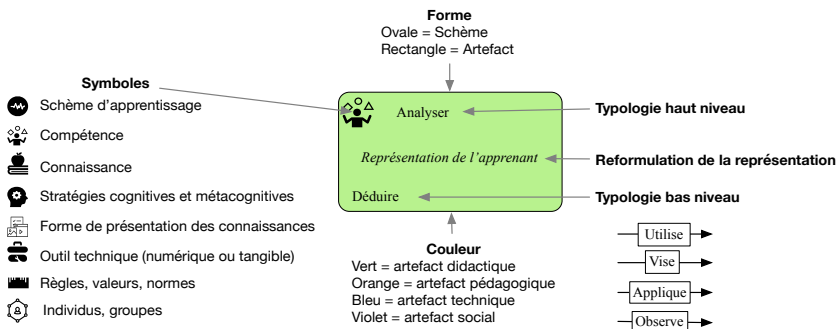


Figure 7 • Système de symboles du langage de modélisation des EPA

Joris FELDER



Figure 8 • Les éléments du langage de modélisation des EPA

6. Application de la méthode MEPA

Nous appliquons ici la méthode MEPA à partir d'un jeu de données collectées (procédure 1) dans le cadre de notre recherche-développement auprès de l'étudiant 13, après 12 semaines de cours de psychologie cognitive du langage. Ce cas est choisi comme exemple pour sa qualité d'illustration, afin d'éclairer la méthode MEPA. Nous décomposons la démonstration de la modélisation en cinq étapes, qui incrémentent tour à tour de nouveaux éléments: le schème et les artefacts techniques, les artefacts pédagogiques, les artefacts didactiques, les artefacts sociaux. Ces étapes mettent en œuvre et répètent les procédures 2 à 4 de la méthode MEPA exposées plus haut. Selon notre expérience, il s'avère en effet pragmatique et efficace de rassembler l'ensemble des données (procédure 1) puis de répéter les procédures 2 à 4 selon les étapes ci-dessous, et, finalement, de procéder à la validation intersubjective de l'ensemble du modèle d'EPA produit. Par ailleurs, nous proposons un commentaire au cours de l'application de cette méthode, pour tenter d'initier quelques réflexions manifestant de l'utilité de la modélisation des EPA.

Verbatim analysé : étudiant 13 (extrait)

Là, c'est sur les stéréotypes et les genres. Je sais plus le nom exact (...). L'article, il y a une étape sur ordi, sur aperçu. Sur l'article même j'ai mis des mots-clés, j'ai souligné. Et après ça je suis revenu sur l'article pendant que je résumais. Je relisais le bout, je résumais, j'allais au prochain. Puisqu'ils [les chapitres] étaient par groupe en fait. Par mots-clés, introduction, méthode. Donc c'était une manière qui allait bien en même temps. Parce que sur l'écran je mettais Aperçu et Word. Et du coup je lisais tout ce que j'avais déjà lu et annoté. Plus je relisais une fois et là j'écrivais tout ce que j'avais besoin d'écrire. (...) Là j'ai fait avec la méthode Cornell. Je ne fais pas toujours comme ça. Mais ça allait bien sur le moment. Je n'ai pas forcément une façon de prendre les notes. (...) Là j'ai pris un cahier exprès pour ça. Pragmatique. J'ai directement mis le nom de l'article, le nom de la personne qui a écrit l'article, ce que je pensais être important. Je mets le nom de l'expérience, le nom des auteurs et puis je décris, enfin vraiment les gros mots-clés qui apparaissent dans l'expérience (...) Le plus rapide si c'est une phrase complète que je ne comprends en anglais pas parce qu'il y a un terme je ne suis pas sûr dedans et que ça peut changer tout le contexte de la phrase, j'utilise Google Translate. Même si ce n'est pas forcément un bon outil de

traduction, généralement ça donne une bonne idée de ce que ça pourrait être. Et puis vraiment si j'ai un gros doute, le dictionnaire anglais-français en ligne. Et on se partage les résumés sur le Google Drive. Il y a des gens qui ne voulaient pas partager les résumés avec ceux qui n'avaient pas participé aux résumés, c'est pour ça qu'il y a eu un Google Drive plutôt que sur la Dropbox où il y a tout le monde qui peut aller chercher.

Étape 1 : identifier un schème et les artefacts techniques (figure 9)

Il faut tout d'abord identifier un schème d'apprentissage en répondant à la question « Que fait l'apprenant pour apprendre ? ».

À partir du verbatim qui précède, le modélisateur évalue à quel niveau de précision il souhaite situer la modélisation. Les données récoltées sont reformulées pour les rendre intelligibles, tout en veillant à rester fidèle aux mots de l'étudiant.

Ainsi, dans la figure 9, l'ellipse réfère au concept de schème d'apprentissage et contient la représentation que s'en fait l'étudiant : « Je lis deux articles. Je les annote. Je fais un résumé. Je partage mon travail avec l'équipe ».

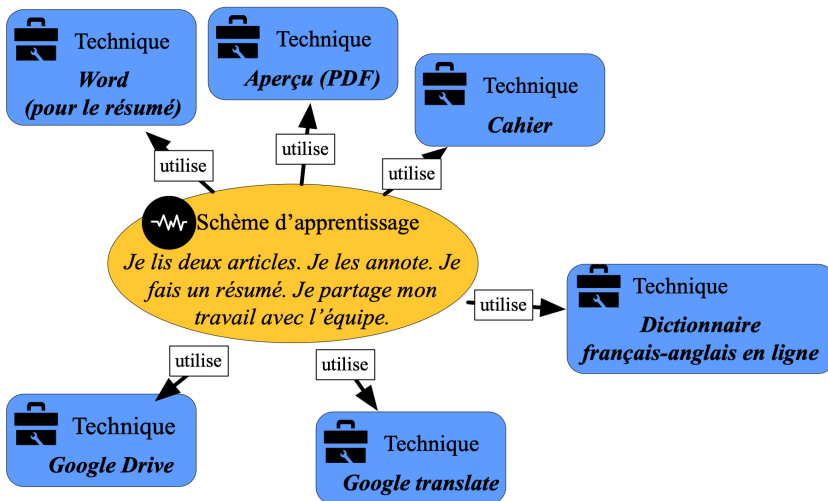


Figure 9 • Modélisation de l'EPA - étape 1

Le schème étant désormais identifié et situé, il devient possible de faire de même avec les artefacts techniques rattachés à ce schème en se posant la question « Qu'est-ce que l'apprenant utilise pour apprendre ? ». Dans notre cas, ce dernier indique qu'il utilise le lecteur de documents Aperçu (Systèmes Mac) pour lire les fichiers PDF, le logiciel Word, Google translate, un cahier, et Google Drive. Ces outils sont représentés dans un rectangle référant au concept d'artefact bleu technique et une flèche avec le label « utilise » est tracée entre le schème et l'artefact. L'ensemble réfère au concept d'instrument.

Étape 2 : identification des artefacts pédagogiques de l'instrument (figure 10)

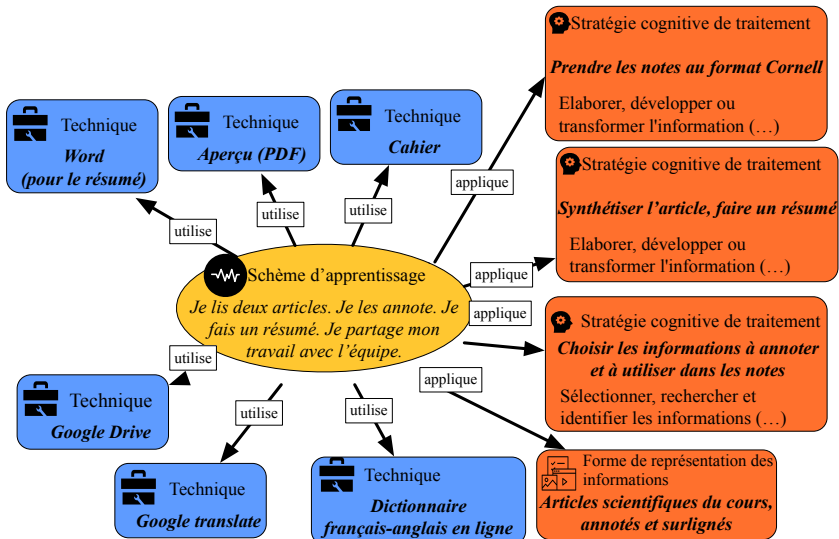


Figure 10 • Modélisation de l'EPA - étape 2

Pour être en mesure de représenter et d'analyser les EPA plus finement, il est nécessaire de dépasser la dimension technologique de l'instrument. Pour ce faire, il faut d'abord se demander « À quelle forme de représentation d'information l'instrument s'applique-t-il ? ». Nous repérons dans ce cas qu'il s'agit d'articles scientifiques fournis dans le cours. Ensuite, on cherche à savoir « Quelle opération mentale l'étudiant applique-t-il ? » pour identifier les stratégies cognitives et métacognitives. Du corpus ci-dessus, nous notons que l'étudiant fait la sélection des informations à

annoter et à surligner ; qu'il prend des notes au format Cornell ; puis qu'il revoit l'article et rédige un résumé. En recourant aux éléments du langage de modélisation, nous pouvons associer ces trois opérations respectivement aux stratégies cognitives de traitement *sélectionner, rechercher et identifier les informations (...)* et *élaborer, développer ou transformer l'information*. Ce faisant, nous constatons que cette dernière stratégie est mise en œuvre par cet apprenant selon deux modalités différentes.

Soulignons que la forme de représentation des informations évolue au cours de l'activité d'apprentissage : ce sont d'abord des articles ; ils deviennent des articles annotés. Finalement, l'apprenant fait un résumé puis prend des notes à partir de son résumé. Si nous avons choisi de modéliser cet ensemble comme un seul instrument, il serait possible de procéder à une modélisation plus fine.

Commentaire : on peut dès lors s'interroger sur la raison menant l'étudiant à entreprendre deux actions de même type et à la pertinence de celle-ci. Est-ce le degré de difficulté de l'article qui l'a poussé à le traiter en deux étapes similaires au niveau cognitif ? Pourrait-il se contenter de traiter l'article scientifique en prenant des notes au format Cornell ? Lui sera-t-il plus aisé par la suite de continuer à apprendre à partir des notes Cornell ou du résumé ?

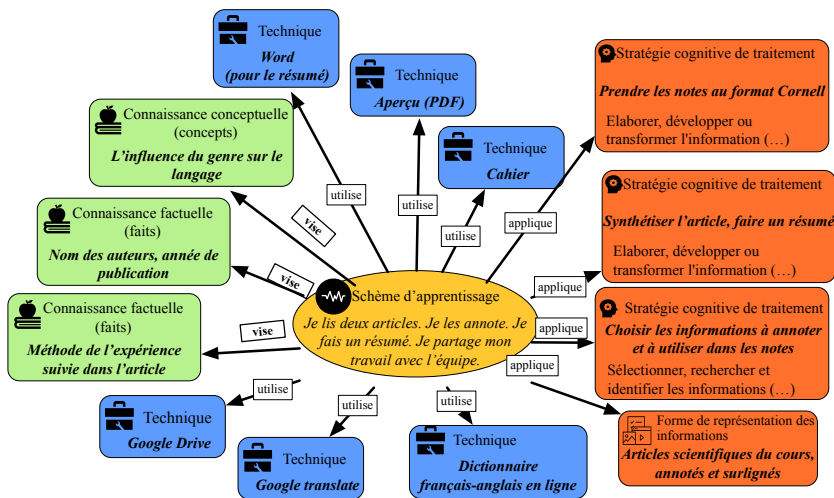


Figure 11 • Modélisation de l'EPA - étape 3

Étape 3 : identification des artefacts didactiques de l'instrument (figure 11)

À cette étape, il faut se demander « Quels objectifs et compétences l'étudiant vise-t-il ? » pour identifier les artefacts didactiques. Du corpus ci-dessus, nous constatons que selon la représentation de l'étudiant, son objectif est d'acquérir des connaissances relatives aux concepts des stéréotypes et du genre³, de retenir le nom de l'auteur et la date de l'article, de comprendre (retracer) le déroulement de l'expérience scientifique relatée dans l'article. En recourant aux éléments du langage de modélisation, nous pouvons associer le premier objectif aux connaissances conceptuelles et les deux derniers aux connaissances factuelles.

Commentaire : cet instrument pourrait-il être modifié afin d'apprendre d'autres concepts et faits ? Comment aider l'étudiant à prendre conscience des compétences qu'il développe ?

Étape 4 : identification des artefacts sociaux de l'instrument (figure 12)

À cette étape, il faut se demander « quels individus ou groupes - quels principes, règles, valeurs externes et personnels - l'étudiant observe ? » pour identifier les artefacts sociaux de l'instrumentation. Dans le corpus de ce cas, nous constatons qu'il apprend en collaboration avec un groupe d'étudiants et qu'il observe le principe d'une répartition équitable du travail tout en se conformant au principe d'informer les membres du groupe en cas de doute quant à la qualité de son travail.

Commentaire : que produirait cet instrument d'apprentissage si l'étudiant n'avait pas pour principe d'informer ses collègues en cas de doute ? Comment l'étudiant s'approprierait-il les connaissances traitées dans des articles préparés par d'autres étudiants ? Serait-il souhaitable, dans le cadre de ce cours précisément, que la collaboration dans les apprentissages prenne la forme d'une répartition entre les étudiants des articles à traiter ?

3 Lors de la validation intersubjective de son modèle, l'étudiant corrige et mentionne que les connaissances visées sont relatives à l'influence du genre sur le langage.

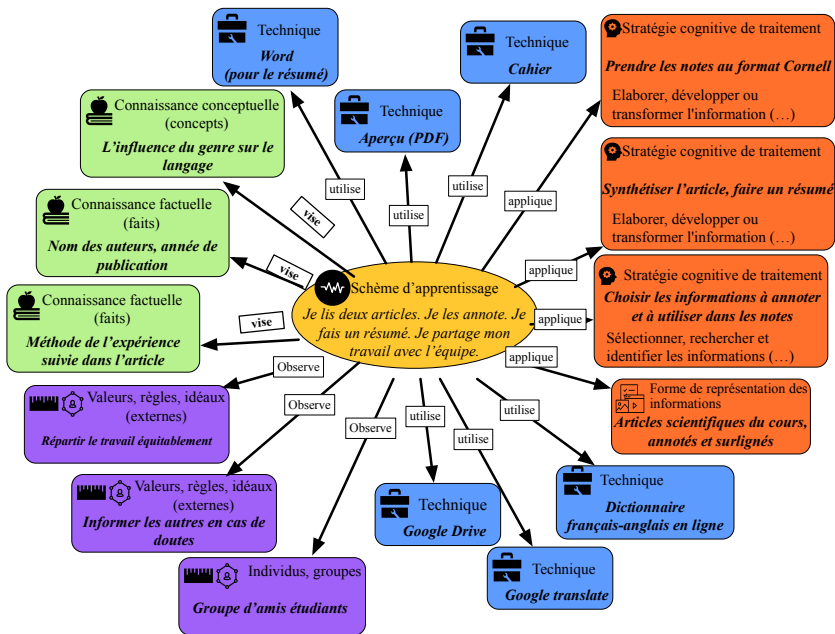


Figure 12 • Modélisation de l'EPA - étape 4

Étape 5 : modélisation de l'EPA complet (figure 13)

Enfin, en analysant l'ensemble du corpus issu de l'entretien avec l'étudiant, nous produisons un modèle de l'EPA représentant l'ensemble des instruments d'apprentissage qui s'organisent en système d'instruments. Certains artefacts techniques utilisés par plusieurs instruments et certaines connaissances (artefacts didactiques) nous apparaissent être visés par l'ensemble de l'instrumentation et non pas par un seul instrument. Pour cette raison nous ne les relierons pas à un schéma en particulier. D'autres instruments complètent celui représenté et décrit précédemment : un instrument dédié à la prise de notes sur le PDF des diapositives du cours lors de la séance en présence, un instrument dédié à la mise en place de la collaboration avec le groupe d'étudiants (communication, partage, organisation), un instrument dédié à l'apprentissage des concepts traités par les autres membres du groupe qui ont fait les résumés d'autres articles scientifiques, un instrument dédié à la préparation à l'examen où l'étudiant identifie la structure attendue et les critères d'évaluation.

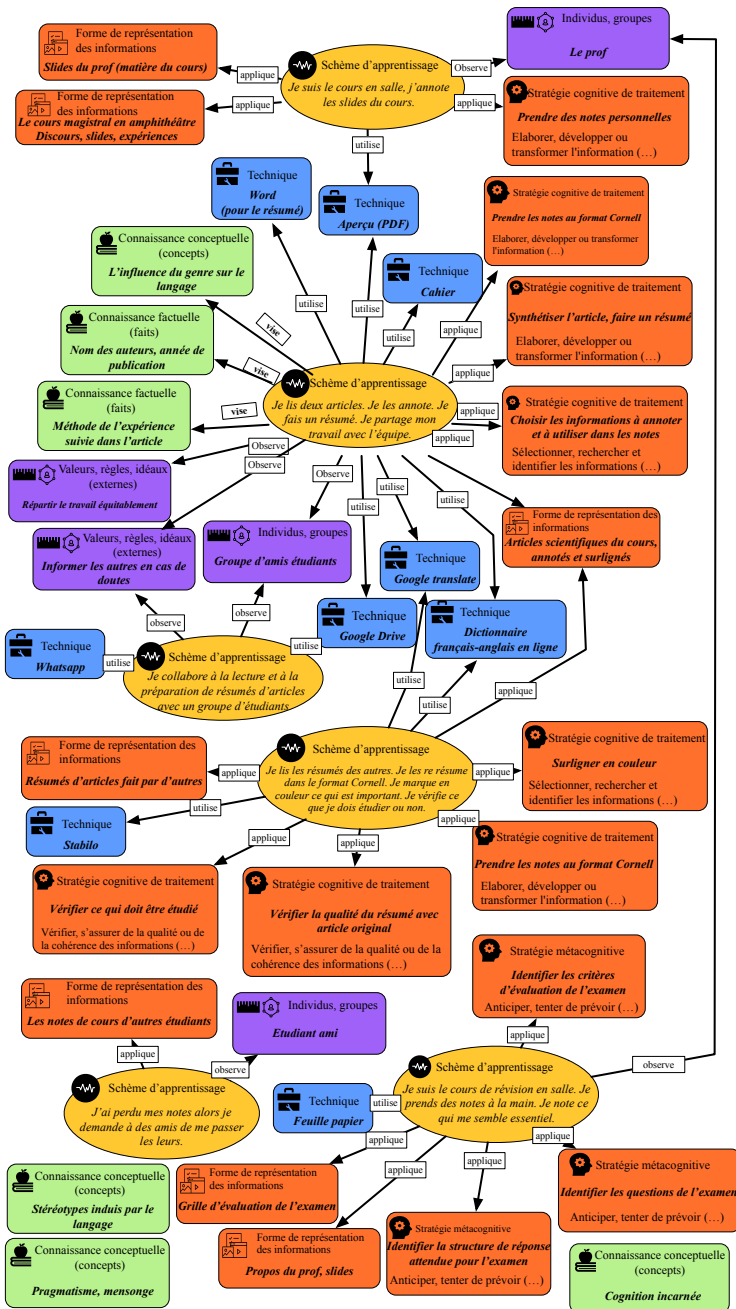


Figure 13 • Modèle de l'EPA de l'étudiant 13, après 12 semaines de cours

Commentaire : on peut dès lors étudier cet EPA dans son ensemble et identifier les fonctions qu'il remplit, ses lacunes, les éventuels conflits instrumentaux (Marquet, 2005), les instruments ou artefacts pivots, les médiations ou encore les catachrèses. Dans le cas présent, l'étudiant aurait-il dû, par exemple, s'entraîner à réaliser un examen blanc ou du moins établir pour lui-même un plan de réponse ? Quelle différence, en termes d'apprentissages, son instrumentation de lecture personnelle des articles produit-elle par rapport à celle où il reçoit les résumés d'articles faits par d'autres étudiants ? La prise de notes en format Cornell lors de la lecture des articles scientifiques ne représente-elle pas un conflit instrumental, dès lors que l'évaluation porte moins sur les connaissances acquises que sur la discussion de la construction des connaissances scientifiques par le biais des expériences de recherche ? Quel rôle particulier jouent les articles scientifiques s'illustrant dans cet EPA clairement comme des artefacts pivots, autour desquels tout le système d'instrumentation s'organise ? Quelle est la relation entre cet artefact pivot et les autres artefacts pivots de cet EPA ? Enfin, en disposant de différents modèles d'EPA à différents moments du cours, il est possible d'étudier les phénomènes d'évolution des EPA.

7. Discussion

Notre objectif était de développer une méthode d'analyse et de modélisation des EPA constituée d'un modèle générique et d'un langage de modélisation de l'EPA. Cet objectif devait répondre aux critères de qualité suivants : expressivité des instances d'EPA figurées par la modélisation, simplicité d'interprétation par les apprenants, portée ontologique des modèles, pragmatisme d'usage comme méthode de recherche et comme outil d'ingénierie pédagogique.

Nous jugeons que le critère d'expressivité des modèles d'instances d'EPA est atteint. En effet, ces modèles reposent sur un langage de modélisation des EPA dont les concepts sont reliés à un modèle générique (sémantique) de l'EPA, valide conceptuellement et empiriquement. Les concepts mobilisés permettent de décrire les EPA avec plusieurs points de vue, augmentant ainsi l'expressivité des informations modélisées (Maisonasse *et al.*, 2010). De plus, ce modèle générique de l'EPA peut servir à l'analyse d'une large variété de situations d'enseignement-apprentissage ainsi qu'à des situations uniques longitudinales. Néanmoins, ce modèle générique est appelé à évoluer avec les apports des futures recherches sur les EPA et sur l'instrumentation de l'apprentissage. Nous

pensons notamment que des propositions de typologies d'outils techniques ainsi que d'artefacts sociaux doivent être développées.

Le critère de simplicité d'interprétation des modèles d'instance des EPA par les apprenants nous paraît être atteint. En effet, lors de la validation intersubjective, aucune difficulté d'interprétation par les étudiants n'est apparue. Certains étudiants anticipaient par moments notre présentation ou nous demandaient d'apporter quelques compléments à leur modèle pour mieux correspondre à leur vécu. Il reste toutefois que ce critère de simplicité d'interprétation devrait être évalué par d'autres études auprès d'étudiants de différents contextes et niveaux de formation.

Nous évaluons le critère de portée ontologique des modèles par leur disposition à permettre l'analyse des propriétés de l'EPA et de ses liens au-delà de leur seule portée représentative. Les modèles produits par la méthode MEPA dépassent largement l'expression morphologique (Maisonasse *et al.*, 2009) des EPA que l'on connaissait jusqu'ici. Ils expriment les EPA comme des systèmes complexes dont les éléments distingués s'assemblent pour former un tout dont le sens est plus fort que la somme des parties (Morin, 2005). Ces modèles nous offrent ainsi des perspectives intéressantes pour l'analyse des phénomènes liés aux EPA. Avec cette instrumentation, le chercheur peut désormais les objectiver et traiter de questions soulevées par la recherche ces dernières années, notamment : quelle est la nature de l'EPA (Henri, 2014) ? Quels impacts les EPA ont-ils sur le devenir personnel et professionnel des personnes (Charlier, 2014) ? Quels sont les impacts des choix collectifs sur les EPA (Peraya et Bonfils, 2014) ? Quelles transformations les EPA engendrent-ils chez l'apprenant ou dans son environnement ? Quelles divergences et tensions les EPA soulèvent-ils (Fluckiger, 2014) ? Quel est le processus de construction et de régulation de l'EPA par l'étudiant (Felder, 2014) ?

À partir de l'ensemble des EPA modélisés dans le cadre de notre recherche, nous observons notamment qu'ils puissent être situés selon quatre dimensions, allant 1) des EPA majoritairement numériques aux EPA minoritairement numériques ; 2) des EPA fortement sociaux aux EPA fortement solitaires ; 3) des EPA présentant une large variété de stratégies cognitives aux EPA à une variété restreinte de stratégies cognitives ; 4) des EPA abondamment instrumentés aux EPA parcimonieusement instrumentés. Ainsi, des réflexions devraient être menées et des recommandations devraient être formulées avec précaution sur le caractère

optimal d'un EPA. S'il est certainement possible d'optimiser un EPA, son équilibre instrumental peut être fragile et son ajustement devrait être réfléchi - à l'image de cet étudiant de notre recherche qui, lors de la discussion de son modèle d'instance a reconnu tout d'abord dans son EPA une grande efficacité en termes de réussite, mais une faible efficacité temporelle et qui a par la suite réfléchi aux possibilités de l'ajuster pour apprendre aussi bien, mais en moins de temps. Il a finalement réalisé que son instrumentation avait aussi pour but de gérer son anxiété lors des examens et il a donc préféré être prudent dans l'ajustement de son instrumentation.

Le critère de pragmatisme d'usage comme référence pour la conduite des recherches de qualité nous semble prometteur. Nous avons en effet pu appliquer rationnellement cette méthode pour récolter et analyser des données issues de 60 entretiens de recherche afin d'en produire autant de modèles d'EPA. Dans le cadre d'une recherche portant sur le potentiel réflexif de ces modèles, nous avons également pu traiter de questions de recherches spécifiques afin d'évaluer les changements conceptuels (Vosniadou, 2007) liés à l'apprentissage auprès d'étudiants universitaires (Felder, 2019). Par conséquent, nous croyons que l'utilisation par les chercheurs d'un même mode de modélisation des EPA serait un vecteur important pour la capitalisation de la recherche. Le modèle générique de l'EPA peut également être étendu pour exprimer les cadres conceptuels mobilisés par d'autres recherches sur les EPA. Le langage OWL (W3C, 2012) et l'outil GMOT (Paquette, 2002) utilisés pour la modélisation du modèle générique permettent pleinement un tel développement par extensions.

8. Conclusion

Cette contribution avait pour objectif de fournir au chercheur et à l'ingénieur pédagogique désireux d'étudier ou d'exploiter les EPA une méthode pour les analyser et les modéliser. Nous avons exposé la méthode de recherche-développement que nous avons suivie pour produire la méthode MEPA, son modèle générique et son langage de modélisation des EPA. Cette instrumentation a été éprouvée dans le cadre d'une recherche longitudinale auprès de 15 étudiants ayant produit 60 modèles d'EPA. Bien que l'évaluation de certains critères de qualité nécessite d'être complétée par d'autres études, le présent article fournit les outils nécessaires pour exploiter la méthode MEPA dans le cadre de

futures recherches. Afin d'en faciliter l'exécution et d'élargir les applications possibles, nous étudions la faisabilité d'un outil numérique.

REMERCIEMENTS

Mes remerciements vont à mes directrices de thèse Bernadette Charlier et France Henri pour leur accompagnement stimulant au long de ce processus de recherche.

RÉFÉRENCES

Aladjem, R. et Nachmias, R. (2013). The mobile as an ad hoc PLE. Learning serendipitously in urban contexts. Dans I. Buchem, G. Attwell, et G. Tur (dir.). *Learning and Diversity in the Cities of the Future. The PLE Conference 2013*, Berlin, 81-88.

Alter, S. (2015). The concept of 'IT artifact' has outlived its usefulness and should be retired now. *Information Systems Journal*, 25(1), 47-60.

Bégin, C. (2008). Les stratégies d'apprentissage : un cadre de référence simplifié. *Revue des sciences de l'éducation*, 34(1), 47-67. <https://doi.org/10.7202/018989ar>

Boutinet, J.-P. (2002). Projet. Dans J. Barus-Michel, E. Enriquez et A. Lévy (dir.) *Vocabulaire de psychosociologie* (p. 222-230). Toulouse, France : ERES. https://www.cairn.info/resume.php?ID_ARTICLE=ERES_BARUS_2002_01_0222

Caron, P.-A., Heutte, J. et Rosselle, M. (2014). Rapport d'Expertise et Accompagnement par la recherche du dispositif expérimental MOOC iNum. <https://halshs.archives-ouvertes.fr/halshs-00950766/document>

Charlier, B. (2017). Enjeu pour la formation des adultes : (re)connaître l'Individu Plus. *Raisons éducatives*, 21, 45-60. <https://doi.org/10.3917/raised.021.0045>

Charlier, B. (2014). Les Environnements Personnels d'Apprentissage : des instruments pour apprendre au-delà des frontières. *STICEF*, 21, 211-237. http://sticef.univ-lemans.fr/num/vol2014/10-charlier-epa/sticef_2014_NS_charlier_10p.pdf

Denis, B. et Joris, N. (2013). *Environnement Personnel d'Apprentissage et Environnement d'Apprentissage Institutionnel Personnalisé : définitions et usages d'étudiants de l'enseignement supérieur*. Communication présentée aux Treizièmes rencontres internationales du réseau de Recherche en Éducation et Formation, Genève. http://sticef.univ-lemans.fr/num/vol2014/11-denis-epa/sticef_2014_NS_denis_11p.pdf

Engeström, Y. (1999). Activity theory and individual and social transformation. Dans Y. Engeström, R. Miettinen et R.-L. Punamäki (dir.), *Perspectives on Activity Theory* (p. 19-38). Cambridge University Press.

Felder, J. (2014). *Construction et régulation de l'environnement personnel d'apprentissage d'étudiants universitaires. Une approche exploratoire* (Mémoire de Master en Sciences de l'éducation). Université de Fribourg, Suisse.

Felder, J. (2017). Comprendre les processus de construction et de régulation des EPA par des étudiants universitaires. *STICEF*, 24(3), 63-95. <http://sticef.univ-lemans.fr/num/vol2017/24.3.3.felder/24.3.3.felder.pdf>

Felder, J. (2019). Le potentiel réflexif de la modélisation des environnements personnels d'apprentissage. *Distance et médiation des savoirs*, 27, 1-21. <http://journals.openedition.org/dms/3962>

Fluckiger C. (2014). L'analyse des environnements personnels d'apprentissage sous l'angle de la discontinuité instrumentale. *STICEF*, 21, 185-210. http://sticef.univ-lemans.fr/num/vol2014/12-fluckiger-epa/sticef_2014_NS_fluckiger_12p.pdf

Guarino, N., Bottazzi, E., Ferrario, R. et Sartor, G. (2012). Open ontology-driven sociotechnical systems: Transparency as a key for business resiliency. Dans M. De Marci, D. Te'eni, V. Albano et S. Za (dir.), *Information Systems: Crossroads for Organization, Management, Accounting and Engineering* (p. 535-542). Physica-Springer. <http://www.loa.istc.cnr.it/wp-content/uploads/2019/10/ITAIS-GuarinoEtAl.pdf>

Henri, F. (2014). Les environnements personnels d'apprentissage, étude d'une thématique de recherche en émergence. *STICEF*, 21, 121-147. http://sticef.univ-lemans.fr/num/vol2014/16-henri-epa/sticef_2014_NS_henri_16p.pdf

Heutte, J. (2014). Persister dans la conception de son environnement personnel d'apprentissage: contributions et complémentarités de trois théories du self (autodétermination, auto-efficacité, autotélisme-flow). *STICEF*, 21, 149-184. http://sticef.univ-lemans.fr/num/vol2014/14-heutte-epa/sticef_2014_NS_heutte_14p.pdf

Hoechsmann, M. et DeWaard, H. (2015). *Définir la politique de littératie numérique et la pratique dans le paysage de l'éducation canadienne*. HabiloMédias. <https://habilomedias.ca/sites/mediasmarts/files/publication-report/full/definir-litteratie-numerique.pdf>

L'Ecuyer, R. (1990). *Méthodologie de l'analyse développementale de contenu. Méthode GPS et concept de soi*. Québec, Canada : Presses de l'Université du Québec.

Mailles-Viard Metz, S., Vayre, E. et Pelissier, C. (2017). Concevoir un Environnement Personnel d'Apprentissage (EPA), est-ce utile pour les étudiants? *Canadian Journal of Education, Canadian Society for the Study of Education*, 38(4). <https://halshs.archives-ouvertes.fr/halshs-01572480/document>

Maisonasse, L., Berrut, C. et Chevallet, J.-P. (2009). L'expressivité des modèles de recherche d'informations précises. Le support de vocabulaires et son application à la recherche d'information médicale. *Document numérique*, 12(1), 107-128. <https://www.cairn.info/revue-document-numerique-2009-1-page-107.htm>

Marquet, P. (2005). Lorsque le développement des TIC et l'évolution des théories de l'apprentissage se croisent. *Savoirs*, 9(3), 105-121. <https://www.cairn.info/revue-savoirs-2005-3-page-105.htm>

Marquet, P. et Leroy, F. (2004). Comment conceptualiser les usages pédagogiques des environnements numériques de travail et d'apprentissage partagés. *Communication présentée à Septième biennale de l'éducation et de la formation*. Lyon, France : INRP. <http://www.inrp.fr/biennale/7biennale/Contrib/longue/3094.pdf>

Martinlade, T. et Dowdy, M. (2016). Issues in Research, Design and Development of Personal Learning Environments. Dans G. Veletsianos (dir.), *Emergence and Innovation in Digital Learning: Foundations and Applications* (p. 119-141). Athabasca University Press. <https://doi.org/10.15215/aupress/9781771991490.01>

Morin, E. (2005). *Introduction à la pensée complexe*. Paris, France : Points.

Nieto, E. et Dondarza, P. (2016). PleS in Primary School: The learners' experience in the Piplep Project. *Digital Education Review*, 29, 45-61.
<https://dialnet.unirioja.es/ejemplar/433604>

Paquette, G. (1996). La modélisation par objets typés – une méthode de représentation pour les systèmes d'apprentissage et d'aide à la tâche. *Sciences et techniques éducatives*, 3(1), 9-42.
https://www.persee.fr/doc/stice_1265-1338_1996_num_3_1_1279

Paquette, G. (2002). *Modélisation des connaissances et des compétences. Un langage graphique pour concevoir et apprendre*. Québec, Canada : Presses de l'Université du Québec.

Peraya, D. et Bonfils, P. (2014). Détournements d'usages et nouvelles pratiques numériques : l'expérience des étudiants d'Ingémédia à l'Université de Toulon. *STICEF*, 21, 239-268. http://sticef.univ-lemans.fr/num/vol2014/19-peraya-epa/sticef_2014_NS_peraya_19.htm

Poizat, G. et Durand, M. (2017). Réinventer le travail et la formation des adultes à l'ère du numérique : état des lieux critique et prospectif. *Raisons éducatives*, 21, 19-44. <https://www.cairn.info/revue-raisons-educatives-2017-1-page-19.htm>

Rabardel, P. (1995). *Les hommes et les technologies. Une approche cognitive des instruments contemporains*. Paris, France : Armand Colin. https://hal.archives-ouvertes.fr/hal-01017462/file/Hommes_et_technologie_Rabardel1995.pdf

Roland, N. et Talbot, L. (2014). L'environnement personnel d'apprentissage : un système hybride d'instruments. *STICEF*, 21, 289-316. http://sticef.univ-lemans.fr/num/vol2014/20-roland-epa/sticef_2014_NS_roland_20p.pdf

Schaffert, S. et Kalz, M. (2010). Persönliche Lernumgebungen: Grundlagen, Möglichkeiten und Herausforderungen eines neuen Konzepts. Dans G. Rainer et P. Otto, *Proceedings of Grundlagen der Weiterbildung - Praxishilfen, Neuwied : Luchterhand, Ergänzungslieferung, Mai 2010*.
<http://www.salzburgresearch.at/publikation/personliche-lernumgebungen-grundlagen-moglichkeiten-und-herausforderungen-eines-neuen-konzepts/>

Trouche, L. (2005). Des artefacts aux instruments, une approche pour guider et intégrer les usages des outils de calcul dans l'enseignement des mathématiques. Dans *Actes du colloque Le calcul sous toutes ses formes* (p. 265-290). France : Académie de Clermont-Ferrand. <https://hal.archives-ouvertes.fr/hal-01559831>

Väljataga, T. et Laanpere, M. (2010). Learner control and personal learning environment: A challenge for instructional design. *Interactive Learning Environments*, 18(3), 277-291. <https://doi.org/10.1080/10494820.2010.500546>

Vartiainen, T. et Tuunanen, T. (2016). Value co-creation and co-destruction in an IS artifact: Contradictions of geocaching. Dans T.X. Bui et R. H. Sprague Jr (dir.), *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)* (p. 1266-1275). IEEE.

Vosniadou, S. (2007). Conceptual Change and Education. *Human Development*, 50, 47-54.

Wilson, M., Scalise, K., et Gochyyev, P. (2015). Rethinking ICT literacy: From computer skills to social network settings. *Thinking Skills and Creativity*, 18, 65-80.

W3C (2012). OWL. Web Ontology Language. <https://www.w3.org/OWL/>



Pensée informatique : points de vue contrastés

► **Béatrice DROT-DELANGE** (INSPÉ Clermont-Auvergne),
Jean-Philippe PELLET (HEP Vaud),
Yannis DELMAS-RIGOUTSOS (Université de Poitiers),
Éric BRUILLARD (EDA, Université de Paris)

■ **RÉSUMÉ** • Issu d'une table ronde organisée lors du colloque Didapro 7 - DidaSTIC à Lausanne en février 2018, ce texte présente différentes analyses contrastées autour de la notion maintenant très populaire de pensée informatique. Il fait le point sur les possibles définitions de la pensée informatique et sur les nombreuses discussions qui lui ont été consacrées. Il adopte trois points de vue : un point de vue historique, un point de vue informatique et un point de vue épistémologique. Le lien avec l'apprentissage de la programmation ou plus largement avec l'apprentissage de l'informatique est également discuté.

■ **MOTS-CLÉS** • pensée informatique, programmation, apprentissage.

■ **ABSTRACT** • *Resulting from a round table organized during the Didapro 7 - DidaSTIC conference in Lausanne in February 2018, this text presents various contrasting analyses around the now very popular notion of computational thinking. It reviews the possible definitions of computational thinking and the many discussions that have taken place about it. It adopts three points of view: a historical point of view, a computer-scientist point of view and an epistemological point of view. The link with learning computer programming or, more broadly, with learning informatics is also discussed.*

■ **KEYWORDS** • *computational thinking, computer programming, computer science.*

1. Introduction

Lors du colloque Didapro 7 - DidaSTIC, qui s'est tenu à Lausanne en février 2018, une table ronde sur la « pensée informatique » a été organisée. Elle a suscité beaucoup d'intérêt parmi les participants et de nombreux échanges ont été amorcés. Comme les actes de la conférence avaient été conçus avant celle-ci (Parriaux *et al.*, 2018), aucun écrit ne rend compte de ce qui a été présenté et discuté, ce qui nous a semblé dommage, notamment au vu des réactions des participants. Au plan international, la notion de *computational thinking* est maintenant abondamment reprise. À titre d'illustration, une page de blog intitulée *Computational Thinking Papers* (<https://cshedresearch.wordpress.com/computational-thinking/>) liste des articles de recherche et des rapports sur cette notion. En se limitant aux articles dont le titre contient le terme *computational thinking*, elle rapporte près de cent articles rien que sur 2018. Côté français, il n'y a pas encore de texte de synthèse autour de cette notion et les références sont rares et éparées. C'est pourquoi nous avons pensé que la table ronde pouvait servir de base à l'élaboration d'un tel texte pour le monde francophone, sur la notion de pensée informatique.

La table ronde posait deux questions, sur lesquelles chaque intervenant a apporté son éclairage. La structure du texte reprend cette trame. La première concernait la notion de pensée informatique, comment la définir et la caractériser. À cette question correspond une partie, constituée par les points de vue des trois participants, que l'on complète par une mise en perspective avec la littérature de recherche en langue anglaise. La seconde interrogeait les implications pour l'enseignement de cette « pensée ». Nous avons décidé de synthétiser les différentes interventions en un texte commun, qui ne distingue plus les différents points de vue, très largement convergents. Enfin, ce texte se termine par quelques éléments prospectifs, tentant d'aller au-delà de la pensée informatique.

2. Qu'est-ce que la pensée informatique ?

Comme nous l'avons mentionné, beaucoup de textes font maintenant référence au *computational thinking* ou à la pensée informatique. Mais comment peut-on la définir ? Cette première section confronte trois éclairages complémentaires : un point de vue plutôt historique porté par Béatrice Drot-Delange, un point de vue d'informaticien détaillé par Jean-Philippe Pellet et enfin un point de vue épistémologique présenté par Yannis Delmas-Rigoutsos.

2.1. Un point de vue historique (Béatrice Drot-Delange)

Les débats, récurrents, sur la nécessité ou non d'introduire l'informatique dans les programmes scolaires mettent en avant des arguments bien connus : considération des besoins de la société pour son industrie liée à l'informatique, renouvellement de l'attractivité des filières scientifiques de l'enseignement supérieur, etc. Ces arguments partagent les caractéristiques de se situer à un niveau macro et à moyen et long terme. D'autres arguments concernent les bienfaits supposés que pourrait apporter cet enseignement aux élèves eux-mêmes, de manière plus ou moins directe. Il s'agit de former les élèves au monde dans lequel ils évoluent et évolueront, en leur donnant les prémices d'une culture informatique et numérique. Les transferts de compétences supposés sont aussi souvent mentionnés, qui s'opéreraient de l'apprentissage des notions informatiques vers d'autres domaines, comme les mathématiques. Les débats sur la pensée informatique relèvent de cette logique de transférabilité.

La question d'une forme de pensée singulière est présente dès les années 1980, lors des débats sur l'introduction d'un enseignement d'informatique en France. Arzac souligne que :

« L'informatique s'appuie sur des méthodes de pensée originales dont l'apport est enrichissant et la valeur culturelle certaine. » (Arsac, 1981).

Faisant l'historique de la méthode informatique, il précise que :

« C'est vers 1965 qu'a commencé à émerger l'idée que l'informatique était plus qu'un phénomène technique, et qu'elle reposait sur une forme de pensée originale qui lui appartenait en propre. Mais c'était alors une intuition plus qu'une évidence tirée des faits. » (ibidem).

En 1983, les orientations décidées par la direction des écoles concernant l'introduction de l'informatique à l'école primaire mentionnent les trois finalités générales de cet enseignement : éveil humain et social, éveil technologique et éveil logistique (Bouille, 1984). Cette dernière finalité met l'accent sur le logiciel et la pensée logistique ou algorithmique.

Un consensus semble s'être élaboré parmi les partisans d'un enseignement de l'informatique au primaire et au secondaire sur l'idée

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

qu'apprendre l'informatique forme les élèves à une pensée singulière liée à la résolution de problèmes. Rogalski insiste ainsi sur l'apport de l'informatique à la résolution de problèmes dans d'autres disciplines grâce à ses outils théoriques (les concepts), les outils techniques renvoyant à l'environnement informatique.

« C'est cet aspect de l'informatique comme outil de pensée qui a été mis en avant dans l'introduction institutionnelle d'un enseignement de l'informatique dans le second cycle des lycées. L'initiation de culture générale à l'informatique implique ainsi une part importante de modélisation de situations-problèmes. » (Rogalski, 1987, p. 7).

Les élèves seraient alors à même de transférer cet apprentissage de la modélisation pour la résolution de problèmes à d'autres domaines. Toutefois, la revue de littérature menée par Baron (1990) souligne que ces transferts de compétences ne sont pas prouvés lors d'expérimentations cherchant à les mesurer.

Dans un article de 2006, Jeannette Wing relance l'intérêt pour la pensée informatique, qu'elle désigne sous l'expression *computational thinking*. Elle précise dans cet article que *« adopter un mode de pensée informatique conduit à résoudre des problèmes, à concevoir des systèmes et à comprendre le comportement humain différemment, en s'appuyant sur les concepts fondamentaux de la discipline informatique et en y incluant une panoplie d'outils intellectuels qui reflètent l'étendue de la science qu'est l'informatique. »* (Wing, 2006).

Elle propose, en guise d'approche de la pensée informatique, une liste d'attitudes et de compétences permises par l'adoption d'une pensée informatique (l'adaptation en français de son texte est disponible en ligne à <https://interstices.info/la-pensee-informatique>). Elle prend soin de distinguer pensée informatique et apprentissage de la programmation, ce qui n'est pas toujours le cas des tenants de la pensée informatique.

« La programmation permet d'implémenter la pensée informatique, mais celle-ci ne s'y réduit pas. (...) Il s'agit bien d'une pensée, pas simplement de calcul mécanique au sens de routinier. (...) La pensée informatique est un moyen mécanique pour les humains de résoudre des problèmes. » (Wing, 2006, p. 35)

Elle reviendra à plusieurs reprises sur ce texte afin de le préciser, l'engouement qu'il a suscité n'ayant d'égal que les interprétations

différentes auxquelles il a donné lieu. Ainsi en 2010, elle précise qu'il s'agit de savoir formuler un problème, pas seulement dans le domaine des mathématiques, mais dans tous les domaines de la vie, qui admette une solution informatique. Il s'agit selon elle de la nouvelle littérature du XXI^e siècle, qui consiste à discerner ce qui relève d'une solution informatique ou non, à choisir les moyens de la mettre en œuvre, à appliquer ou à adapter les outils ou techniques à de nouveaux usages, etc. En 2014, elle insiste en disant que la pensée informatique ne concerne pas seulement la résolution d'un problème mais aussi sa formulation. On peut faire le lien avec l'idée de modélisation présente tant chez Arzac que chez Rogalski. Elle indique également qu'il s'agit de penser comme un informaticien. La pensée informatique comme étant celle des informaticiens est également reprise par le groupe de travail européen CompuThink (<https://ec.europa.eu/jrc/en/computational-thinking>). La pensée informatique est, selon ce groupe, l'utilisation des concepts de l'informatique pour formuler et résoudre un problème.

La pensée informatique ainsi définie, celle des informaticiens, parfois limitée aux programmeurs, est-elle aussi l'approche proposée par Papert ? Papert prône l'importance de l'apprentissage d'un langage de programmation, comme moyen d'enseigner aux enfants à penser :

« [...] certaines utilisations d'une technique informatique puissante, voire d'un état d'esprit informatique, peuvent fournir aux enfants de nouvelles façons d'apprendre, de penser [...] » (Papert, 1981, p. 30).

Mais son propos concerne bien davantage les idées fondamentales que les enfants peuvent découvrir ou s'approprier dans des situations informatisées que l'apprentissage de l'informatique pour elle-même (Papert, 2000). Bers (2017) revient sur les trois rôles que peuvent jouer les ordinateurs en lien avec des idées fondamentales selon Papert.

- Un rôle neutre : certaines idées fondamentales sont indépendantes de l'existence de l'ordinateur, elles existaient bien avant l'informatique et celle-ci n'entraîne pas de modification majeure sur ces idées.

- Un rôle libérateur : certaines idées existaient avant l'informatique, mais l'ordinateur les libère en les rendant plus puissantes et plus accessibles à un grand nombre. La modélisation en serait un exemple.

- Un rôle incubateur : un petit sous-ensemble d'idées sont nées avec l'ordinateur, elles ne pourraient pas être connues sans l'ordinateur.

Lors de sa conférence au colloque Didapro 7 - DidaSTIC, Bers (2018) indique la dimension protéiforme de la pensée informatique, dans la

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

lignée des travaux de Papert : pensée abstraite, systémique, logique et séquentielle, algorithmique, résolution de problèmes, apprentissage par l'erreur... Il ne s'agit pas seulement d'apprendre à faire quelque chose, mais aussi d'aborder différemment des concepts ou notions, d'exprimer des idées, et finalement de s'exprimer. Elle est à l'origine d'environnements (ScratchJr et Kibo) qui visent à permettre aux enfants dès le plus jeune âge d'être créatifs, de s'exprimer et de découvrir des idées fondamentales (Bers, 2017).

Le rapide historique, non exhaustif, montre les enjeux autour de la notion de pensée informatique, comme légitimation d'un enseignement de l'informatique dans le système scolaire. Les travaux rapidement mentionnés renvoient à la question de la spécificité et de la caractérisation de la pensée informatique par rapport à d'autres formes de pensée.

2.2. Un point de vue informatique (Jean-Philippe Pellet)

Pour toute communauté professionnelle ou scientifique, c'est à la fois une aubaine et un risque de voir se développer un intérêt massif pour ce qui constitue ses thématiques et ses problématiques propres. C'est une aubaine, parce que c'est l'occasion de faire mieux connaître la discipline. Il y a tout à coup des leviers à actionner : politiques, financiers, collaboratifs. On a l'occasion de parler de notre milieu, de notre passion, ceci en ayant tout à coup une considération du grand public différente et une pertinence nouvelle. Et d'un autre côté, un tel intérêt est aussi un risque — de l'intérêt massif soudain s'ensuit un discours public parfois dépourvu de la connaissance des notions et concepts des professionnels.

Par le côté « transfert » relatif aux compétences que la pensée informatique semble désigner, par la pluridisciplinarité qu'elle reflète, ou encore par la position assumée de, par exemple, Jeannette Wing (2006) de la présenter comme universellement applicable et utile, il suit que chacun se sent concerné par une discussion autour de la pensée informatique. Il est d'autant plus important d'essayer de cerner les interprétations qui, au-delà du *buzzword*, sont les plus à même d'apporter aux étudiants quelque chose d'aussi fondé, stable et enrichissant que possible.

Le premier constat à faire, même s'il est peut-être évident, est que la « pensée informatique », telle qu'on la décrit aujourd'hui, n'est pas un concept informatique. C'est un concept lié à l'enseignement de l'informatique (et de thématiques connexes), à sa potentielle transversalité, aux

compétences générales développées, et aux potentiels transferts que sa maîtrise ou sa connaissance impliquerait. On pourrait s'aventurer à dire que c'est d'autant plus un thème lorsque l'on parle d'un enseignement destiné à un public général et non à de futurs informaticiens, où la nécessité d'« enrober » des connaissances ou compétences d'un nouveau vocabulaire semble moins présente.

En tant qu'informaticien, la lecture la plus structurante sur la pensée informatique/*computational thinking* est probablement l'article de Peter Denning, « *Remaining Trouble Spots with Computational Thinking* » (Denning, 2017). Denning, qui a été président de la grande *Association for Computing Machinery*, y pose notamment trois questions, relayées brièvement ici.

1. Qu'est-ce que le *computational thinking* ? Denning montre que le concept ne date pas de Jeannette Wing en 2006, mais, comme évoqué ici plus haut, des années 1960 et plus concrètement encore du virage computationnel des années 1970 et 1980. C'est à ce moment que la simulation informatique est devenue un troisième grand moyen de faire des découvertes scientifiques, aux côtés de l'approche théorique et de l'approche expérimentale. Il explique aussi pourquoi il pense que c'est une erreur de déconnecter des savoir-faire computationnels du modèle d'exécution qu'il y a derrière, souvent d'une machine, comme le fait Wing : parce que c'est dans le but de faire accomplir un certain travail au modèle que l'on met en œuvre ces savoir-faire.

2. Comment mesurer des apprentissages en *computational thinking* ? Il dit que les tests portent traditionnellement sur des savoirs et pas des savoir-faire et que cela pose problème : en effet, la définition de *computational thinking* de Jeannette Wing mentionne presque exclusivement des savoir-faire plutôt que des savoirs.

3. Le *computational thinking* serait-il absolument utile pour tout le monde ? Denning le conteste et se déclare sceptique en ce qui concerne le transfert des compétences. Il argumente que la plus-value de la maîtrise des compétences derrière le *computational thinking* n'est vraiment réelle que pour les gens qui doivent effectivement concevoir des calculs automatiques, de la *computation*. Ceci ne veut en aucun cas dire que le *computational thinking* ne serait pas pour tout le monde – juste qu'il ne faut pas s'aventurer à des promesses irréalistes, générer des attentes démesurées pour ensuite ne pas être en mesure d'y répondre en conséquence.

Ceci dit, il y a bien un avant et un après Wing. Il est parfois difficile de bien cerner le changement de focale que cela provoque sur l'enseignement de l'informatique (ou, d'ailleurs, d'autres disciplines) et la cause de son succès. Dans ce sens, Denning compare la pensée informatique traditionnelle avec la « nouvelle » pensée informatique de Wing et pose la différence d'approche fondamentale qu'il y voit : dans le contexte de la pensée informatique traditionnelle, c'est la pratique de la conception d'algorithmes et de la programmation qui engendre du *computational thinking*, alors que, dans le contexte de la nouvelle pensée informatique, c'est l'apprentissage de certains concepts transversaux qui aident à la résolution de beaucoup de problèmes – dont la programmation, qui n'en est qu'un parmi d'autres. La direction de la causalité est inversée.

Pourquoi l'approche de Wing a-t-elle trouvé autant d'écho cette dernière décennie ? Peut-être qu'une partie de réponse réside dans cette idée en arrière-plan, séduisante mais, à mon avis, trompeuse, qu'on pourrait y trouver, à savoir : il est inutile d'acquérir des connaissances et une pratique disciplinaire importantes pour maîtriser les compétences de plus haut niveau inhérentes à la discipline et pour discuter de leur pertinence dans la formation générale.

2.3. Un point de vue épistémologique (Yannis Delmas-Rigoutsos)

Qu'est-ce que la « pensée informatique » ? Est-ce la pensée des informaticiens ? Lesquels, dans ce cas ? Celle des créateurs de logiciels, qu'on a pu appeler « pensée logistiquette » ? Celle des scientifiques du domaine informatique ? Les présentations du colloque Didapro 7 - DidaSTIC (Parriaux *et al.*, 2018) montrent combien les points de vue sont, en réalité, divers. Certains envisagent de mettre au centre la résolution de problèmes – c'est très souvent évoqué. Pourquoi pas, mais cette démarche est très loin d'être spécifique à l'informatique. C'est un élément important de la culture des ingénieurs, dont participent nombre d'ingénieurs informaticiens. On pourrait aussi s'attacher à ce qui en découle souvent : l'utilisation de « boîtes à outils » (outils technologiques ou intellectuels).

À notre sens, il ne faut pas espérer pouvoir disposer d'une définition autre qu'en extension de la pensée informatique. L'histoire de l'informatique, y compris pour ses principaux concepts, est nourrie de notions qui ne lui sont pas spécifiques et qui, pour certaines, préexistent de beaucoup : algorithme, programme enregistré, arbitraire du signe linguistique... C'est d'ailleurs ce qu'on retrouve dans l'histoire de nombreuses autres sciences

et techniques : à certains moments convergent de nombreux apports, divers et parfois anciens. Dans le cas de l'informatique, soulignons l'apport d'inspiration de la littérature, en particulier de science-fiction.

C'est une illusion totale de croire qu'on pourrait avoir une définition en intension de l'informatique — là encore, l'informatique ne se distingue pas de nombreuses autres sciences. Une discipline scientifique est un patchwork parce que son histoire, l'histoire des apports qui l'ont construite, est elle-même un patchwork.

Désigner tel outil (intellectuel ou technologique) comme « important » relève d'un choix (Delmas-Rigoutsos, 2018). En particulier ce choix ne peut que dépendre de la destination de cet outil : « important » n'a guère de sens en soi ; il convient de toujours préciser important *pour tel public* (ou pour telle finalité). Pour cette raison, le cœur de ce qu'est la pensée informatique varie en fonction du public visé. La notion de « pensée informatique » admet inévitablement différentes versions, ou plus exactement instanciations.

Pour Denning (2003), il faudrait nécessairement retenir comme centrale l'idée du modèle computationnel, au sens de modèle du calcul. Or l'informatique comporte également de nombreuses autres contributions à la formation générale de l'esprit, et notamment le rapport à l'abstraction, à la structuration des faits, à la représentation des objets de la pensée. Peter Denley, cofondateur et premier Secrétaire général de l'Association for history and computing, explique que comprendre le principe d'organisation d'une base de données relationnelle serait, pour tous les historiens, « *a considerable service to their understanding of structured systems of any kind* » (Denley, 1994, p. 34). Plusieurs projets de recherche en histoire ont été l'occasion de l'observer, de manière très concrète. La représentation des données n'était pas pensée computationnellement : de fait, les projets utilisaient leurs bases de données comme on aurait pu le faire quelques années plus tôt avec des fiches cartonnées (avec, certes, le confort d'une forme de traitement de texte). Plusieurs de ces projets de recherche ont périclité, ou échoué à se développer, du fait d'un manque de conceptualisation de leurs jeux de données.

Revenons à la question de la définition de la pensée informatique. Bien sûr, les représentations évoquées sont souvent dirigées (ou devraient l'être) par la possibilité d'opérer des traitements sur les données, et, en ce sens, elles sont donc bien computationnelles, mais il ne s'agit pas là du tout, pour autant, de modéliser le calcul lui-même. D'ailleurs, si l'on

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

observe la manifestation de cette compétence dans d'autres sciences, il s'agit bien souvent plus de comprendre des phénomènes que de traiter des données, *stricto sensu*. La dimension de l'analyse computationnelle (c'est-à-dire les techniques de recherche d'une représentation adaptée, de construction d'un référentiel) peut être un apport important à de nombreuses autres disciplines scientifiques.

Nous pourrions citer d'autres exemples de cette dimension computationnelle, notamment la simulation. Il semble que ce serait trop réducteur de ne voir là que des outils techniques de calcul. La simulation est aussi un apport intellectuel de l'informatique, ne serait-ce que parce qu'elle nous oblige à préciser les variables de modélisation, les éléments à abstraire, voire à négliger...

L'expression « pensée informatique » a peut-être comme intérêt essentiel, au-delà de son contenu, d'insister sur l'idée qu'il s'agit en propre d'une pensée et non seulement de savoir-faire technologiques et, ce faisant, de promouvoir l'articulation entre l'informatique et les autres disciplines, notamment scientifiques. Dans une des présentations au colloque (Chessel Lazzarotto, 2018), un réemploi en grammaire d'une notion vue en robotique est un exemple, parmi de très nombreux autres, qui montre l'intérêt d'une telle démarche.

De nombreux liens intellectuels, épistémologiques, seraient intéressants à développer. Il y aurait à faire, par exemple, du côté de la cybernétique, qui rentre dans le mouvement intellectuel plus large du structuralisme, qui va des mathématiques avec Bourbaki à l'anthropologie avec Claude Lévi-Strauss, et qui est également au cœur de l'informatique.

Autre exemple : la correspondance preuve-programme (type-calcul), qui a un lien très fort avec l'analyse dimensionnelle en physique, ou avec la combinatoire énumérative en mathématiques. Celle-ci pourrait alimenter une réflexion sur l'apprentissage élémentaire du calcul – peut-être même fonder ces apprentissages. De tels échanges, en approfondissant la correspondance de Curry-Howard, permettent d'ailleurs de mieux comprendre certaines notions ou méthodes de l'informatique, en les généralisant et en les explicitant, par exemple la nécessité pour un code d'être clairement documenté : un code, c'est au moins autant des instructions que du typage, de la spécification, de la documentation. Ce serait, enfin, une occasion de valoriser la dimension expressive, poétique, artistique, du code.

Si l'on élargit un peu le propos, il peut être intéressant d'observer l'informatique comme une matrice disciplinaire, telle que la formalise l'épistémologue Imre Lakatos. Pour cet auteur, les programmes de recherche scientifiques se structurent dans deux directions : d'une part une heuristique¹ positive indique dans quel sens chercher, tandis qu'une heuristique négative correspond à la base théorique (méthodes, vocabulaire, concepts, théories...) sur laquelle s'entendent les participants de la discipline, par large consensus, et qu'ils souhaitent, pour l'heure, ne pas remettre en question. De ce point de vue, si l'on s'intéresse à une notion de pensée informatique à très gros grain ou pour des non-spécialistes, on peut rester, pour l'essentiel, dans le champ de l'heuristique négative : la discipline est stable, bien délimitée, et, pour l'essentiel, assez explicitable. Si, en revanche, on s'intéresse à la science en marche, à la recherche vivante, le champ s'étend à l'heuristique positive, par définition protéiforme et mouvante. À ce niveau, il ne faut pas, d'une décennie à la suivante, espérer trouver les mêmes composantes d'une hypothétique pensée informatique : les contours de la discipline changent perpétuellement. Ce qui, à une époque, semble important peut devenir plus tard secondaire. Si l'on veut poser la question dans le champ de l'éducation, il devient donc essentiel de poser des objectifs pédagogiques : dans quelles directions, avec quelles intentions, souhaite-t-on travailler la pensée informatique ?

2.4. Pensée informatique et *computational thinking* : analyse de revues de questions en langue anglaise

Les questionnements qui viennent d'être évoqués trouvent des échos dans les nombreuses revues de littérature en langue anglaise consacrées au *computational thinking*. Nous en donnons un aperçu ci-après.

Plusieurs revues ont été publiées depuis 2013. On peut citer :

- Grover et Pea, pour l'enseignement primaire et secondaire (Grover et Pea, 2013) ;
- Kalelioglu, Gülbahar et Kukul, s'intéressant à établir un cadre pour la pensée informatique (Kalelioglu *et al.*, 2016) ;
- Shute, Sun et Asbell-Clarke, se proposant de démystifier le *computational thinking* (Shute *et al.*, 2017) ;

1 Le mot *heuristique* s'entend ici au sens de : mouvement de découverte.

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

- Hickmott, Prieto-Rodriguez et Holmes, explorant le lien avec l'apprentissage des mathématiques (Hickmott *et al.*, 2017) ;
- Hsu, Chang et Hung, se centrant sur des questions d'apprentissage et d'enseignement (Hsu *et al.*, 2018) ;
- Lockwood et Mooney, pour l'enseignement secondaire (Lockwood et Mooney, 2018) ;
- Kirwan, Costello et Donlon, se limitant à l'apprentissage en ligne (Kirwan *et al.*, 2018).

On peut également citer un numéro spécial de la revue *International Journal of Child-Computer Interaction* consacré à la pensée informatique et à l'apprentissage du code pour les jeunes enfants (Howland *et al.*, 2018).

Des recherches visent à circonscrire la pensée informatique, d'une part, comme une somme de concepts ou de compétences. C'est le cas par exemple des travaux de Csizmadia *et al.* (2015), pour qui la pensée informatique recouvre les compétences d'abstraction, de généralisation, d'évaluation, de pensée algorithmique et de décomposition. Malgré tout, ces compétences très générales restent difficilement observables lors de l'activité des élèves (Drot-Delange et Tort, 2018).

D'autre part, des recherches s'appuient sur les travaux existants pour tenter de repérer les invariants et évaluer le consensus des experts sur telle ou telle conception de la pensée informatique. C'est le cas de l'approche présentée dans (Selby et Woollard, 2013). Les auteurs constatent le nombre croissant de chercheurs ou d'acteurs du monde éducatif mobilisant la notion de pensée informatique. Ils mènent une revue de littérature pour retenir les concepts associés à la pensée informatique qui font consensus et ont une définition robuste.

Trois concepts semblent répondre à ces critères : processus de pensée, abstraction et décomposition. Les auteurs retiennent ensuite les concepts suivants, qui présentent cependant moins de constance que les précédents : pensée algorithmique, évaluation et généralisation.

Il est intéressant aussi de noter ce qu'ils ne retiennent pas, la pensée logique ou d'autres termes similaires : pensée mathématique, pensée heuristique en sont des exemples... Pourtant certains auteurs les incluent dans la pensée informatique, en considérant qu'elle serait l'équivalent du raisonnement logique effectué par les humains, à l'image de l'intelligence artificielle ou des activités proposées par le courant de l'informatique débranchée (Drot-Delange, 2013). S'ils décident de ne pas retenir cette

notion de pensée logique, c'est qu'elle est selon eux sujette à des interprétations trop ouvertes, qui empruntent à d'autres disciplines telles que les sciences de l'ingénieur ou les mathématiques. Selby et Woollard (2013) considèrent que ces emprunts n'aident pas vraiment à définir la pensée informatique.

Le manque de consensus sur la notion même de pensée informatique est un constat largement partagé. Comme le remarquent Shute et ses collègues (Shute *et al.*, 2017), la définition même évolue avec l'accumulation des connaissances sur cette pensée informatique. La multiplicité de ces revues de la littérature atteste que son opérationnalisation ne va pas de soi, montrant la difficulté d'une approche finalement plutôt abstraite et théorique. D'ailleurs, Grover et Pea (2013) se demandent pourquoi l'article princeps de Wing (2006) a eu tant de succès et a servi de cri de ralliement (« *rallying cry* ») aux éducateurs, chercheurs en éducation et aux administrateurs. La pensée informatique serait une sorte d'étendard et il conviendrait maintenant d'analyser ce que font ceux qui s'y rallient. Une démarche ascendante est ainsi privilégiée : il ne s'agit pas de savoir si la notion est correctement appliquée, mais d'étudier comment ce qui est fait concourt à en préciser la définition.

S'agissant des références, les différents auteurs cités s'accordent pour remonter aux travaux de Seymour Papert, soulignant le côté enseignement et apprentissage de la pensée informatique, puis à l'article de Jeannette Wing de 2006. L'apprentissage par le jeu (*game-based learning*) et le constructivisme sont les théories principales qui servent de base aux articles sur la pensée informatique (Kalelioglu *et al.*, 2016 ; Kirwan *et al.*, 2018).

On peut noter l'intérêt des grandes associations américaines, comme CSTA (*Computer Science Teachers Association*) et ISTE (*International Society for Technology in Education*), et aussi des grandes entreprises telles que Google, avec le site *Exploring Computational Thinking*, (<https://edu.google.com/resources/programs/exploring-computational-thinking>) qui se veut un support pour les enseignants qui souhaitent intégrer la pensée informatique dans leurs pratiques d'enseignement), ou Microsoft qui propose une formation sur la pensée informatique et son utilisation dans différents contextes éducatifs sur le site *Computational Thinking and its importance in education* (<https://preview.education.microsoft.com/en-us/course/a41b9507/overview>). Notons également l'article de Barr et

Stephenson (2011) qui a proposé une définition opératoire de la pensée informatique.

L'importance de la pluridisciplinarité est souvent mise en exergue et les champs concernés par la pensée informatique sont nombreux. Ainsi, Mazzone (2018) voit en Andy Warhol un modèle pour la pensée informatique et la création artistique, dans la génération et la production de formes, tant en peinture que sur les premiers ordinateurs Amiga : une simulation de l'abstraction des processus et des méthodes de production qui nous sont familiers dans l'art électronique computationnel d'aujourd'hui.

L'utilisation de concours informatique, comme le concours Castor (Drot-Delange et Tort, 2018), est mentionnée comme un vecteur important pour développer la pensée informatique. La question de la transférabilité, toujours complexe, est souvent évoquée. Mais plutôt que parler de transfert, on peut remarquer que l'informatique, avec ses instruments et ses formes de travail, s'est imposée dans les recherches au cœur de la plupart des disciplines. C'est un mode de pensée, via des instrumentations spécifiques, qui a été repris et développé dans de nombreux champs disciplinaires.

Enfin, dans les mises en œuvre de la pensée informatique, beaucoup regrettent le manque de recherches expérimentales ou quasi-expérimentales dans les classes ainsi que la rareté des évaluations.

3. Quels choix de formation ?

Comme nous venons de le voir, la notion de pensée informatique est sujette à caution, mais, même si chacun la redéfinit partiellement selon ses propres filtres, elle est largement utilisée. Peut-être que cette notion, victime de son succès, a perdu sa signification avec la généralisation de son emploi. En tous cas, cela montre l'intérêt de discuter de son régime de fonctionnement pour la formation.

Pour ce faire, un jeu à trois termes se présente (informatique, pensée informatique et programmation), avec des définitions de la pensée informatique plutôt fluctuantes qui s'inscrivent dans les objectifs ou finalités des formations.

3.1. La pensée informatique et l'informatique ne se résument pas à la programmation

Le fait que l'informatique et la pensée informatique ne se résument pas à la programmation semble bien admis. La pensée informatique pourrait d'ailleurs presque s'en passer. Ainsi, dans le cadre d'une étude européenne (Bocconi *et al.*, 2016), il est constaté que certains chercheurs considèrent qu'il n'y a pas nécessité, pour développer la pensée informatique, de programmer un ordinateur. Ce résultat pourrait s'obtenir en mettant en œuvre une démarche de résolution de problèmes mobilisant des stratégies comme des algorithmes, l'abstraction ou le débogage. Toutefois, pour d'autres, la programmation est une étape incontournable, car c'est l'apprentissage des concepts centraux en informatique qui permettront de développer cette forme de pensée (Voogt *et al.*, 2015).

S'il est admis que pensée informatique et programmation sont distinctes, les travaux de recherche sur la pensée informatique prennent le plus souvent comme appui des contextes de programmation. Ceci pourrait conforter l'idée que c'est la même chose, ou qu'au moins il est nécessaire de programmer pour développer une pensée informatique. Ainsi certaines activités, comme la programmation en Scratch, permettraient de développer la pensée informatique. Cette idée est présente dans le rapport de l'Académie des sciences (2013), dans lequel il est mentionné que « l'apprentissage de la programmation permet de découvrir les rudiments de la pensée informatique » (p. 24). Mais rares sont les recherches qui visent à produire des résultats sur ces liens.

S'agissant d'informatique, l'accord semble se faire sur le rôle forcément limité, même s'il est essentiel, de la programmation. Il est possible de faire de la musique sans faire de solfège, mais il est (presque) impossible d'en faire sans jouer d'un instrument. Il semble que le parallèle existe pour l'informatique : on peut faire de l'informatique sans apprendre de formalisme, au moins dans un premier temps, mais il n'est pas possible d'en faire sans passer, d'une façon ou d'une autre, par une activité de programmation.

Pour autant, Jeff Atwood, un grand nom de l'informatique appliquée – fondateur du site site Stack Overflow (<https://stackoverflow.com>) – dans un article intitulé « *Please don't learn to code* » (Atwood, 2012), insiste sur l'erreur qui consiste à considérer la programmation comme une fin en soi. À la place, il recommande de rechercher avant tout à comprendre

comment les choses autour de nous fonctionnent à leur niveau élémentaire (et à mieux communiquer avec les autres êtres humains).

Mais comment expliciter la partie de l'informatique qui n'est pas programmation ?

3.2. L'informatique ne se résume pas à la pensée informatique

Cette question conduit à faire ressortir un des enjeux de l'enseignement de l'informatique. Il ne s'agit pas « seulement » de développer des compétences plus ou moins abstraites et transférables de résolution de problèmes ; il s'agit aussi d'expliquer le fonctionnement du monde qui nous entoure.

En ce sens, mettre de côté « informatique » pour garder uniquement « pensée informatique » nie ces enjeux. Une approche où une forme de pensée informatique serait enseignée de façon complètement déconnectée de l'informatique elle-même ne répondrait pas au besoin de formation de futurs citoyens capables de s'exprimer de façon fondée sur des enjeux sociétaux potentiellement techniques et complexes.

Pour faire une comparaison avec l'enseignement des sciences naturelles, il y a bien sûr, derrière cet enseignement, une volonté de parler de la démarche scientifique en général et de cette façon structurée de construire des connaissances et de faire des découvertes. Ce n'est pas sans parallèle avec un enseignement de l'informatique qui viserait à montrer comment des problèmes complexes (organisation d'immenses masses de données, gestion de multiples processus en parallèle, ou méthodes d'apprentissage automatique) ont été approchés, décomposés et traités par les informaticiens – enseignement pendant lequel on peut spécifiquement s'attarder sur des techniques générales de résolution de problèmes particulièrement opportunes. Mais tout comme la démarche scientifique traitée hors discipline scientifique semble largement désincarnée, on peut considérer que la pensée informatique sans l'informatique est privée de sa « substantifique moelle ».

Faire le lien avec les questions scientifiques en arrière-plan est fondamental pour donner du sens aux activités et aux apprentissages. C'est d'ailleurs cette approche qu'adopte le concours Castor, où chaque question est agrémentée d'un paragraphe « C'est de l'informatique » qui explique comment les stratégies mises en œuvre dans la résolution de la

question sont similaires à celles mises en œuvre dans de réelles problématiques informatiques.

Notons toutefois que le *computational thinking* est présenté explicitement dès l'introduction des programmes en informatique du Royaume-Uni comme un objectif de cet enseignement pour comprendre et changer le monde (*Department for Education, 2013*). L'élaboration de la relation entre pensée informatique et items du programme reste à la charge de l'enseignant (*Drot-Delange et Tort, 2018*).

3.3. La pensée informatique accompagne l'introduction de l'informatique dans les programmes

Ainsi, ce sont les modalités d'introduction de l'informatique dans les différents systèmes éducatifs qu'il convient d'observer. Pour les pays nordiques, Bocconi, Chiocciariello et Earp (2018) décrivent une approche associant programmation et pensée informatique. Dans (*Heintz et al., 2016*), les auteurs étudient comment dix pays différents ont abordé la question de l'introduction de l'informatique dans leur éducation primaire et secondaire. Selon eux, la pensée informatique est rarement mentionnée explicitement, mais les idées associées sont souvent incluses sous une forme ou une autre.

Il semble que l'intérêt de la notion de pensée informatique est d'amener à réfléchir aux finalités à donner à un enseignement de l'informatique dans un contexte scolaire. Une piste prometteuse semble être la réflexion sur les idées fondamentales à l'aide des critères proposés par Bers (2017) : une idée est fondamentale, selon elle, si elle personnellement utile à l'apprenant, si elle est épistémologiquement liée à d'autres disciplines, si elle trouve ses racines dans les connaissances intuitives qu'un enfant a pu développer sur une période longue.

Si on retrouve certaines dimensions développées par Schwill (*Hartmann et al., 2012*), Bers introduit le critère de l'intérêt pour l'individu lui-même, certes décrété *a priori*, et ses connaissances personnelles, comme constitutif du caractère fondamental d'une idée. Cela semble constituer une passerelle intéressante avec les développements en didactique de l'informatique, comme ceux de l'informatique « située » (*Guzdial, 2010* ; (*Knobelsdorf et Tenenber, 2013*), ou le modèle de reconstruction didactique de l'enseignement de l'informatique (*Bruillard, 2017*), ancrés dans les pratiques informatiques quotidiennes des individus.

3.4. Des choix de formation prenant en compte plusieurs enjeux

Finalement, l'implication la plus claire de la pensée informatique sur l'enseignement de l'informatique est d'en repenser les enjeux et les modalités. Sur ce dernier point, toutefois, la pédagogie de projet semble s'imposer dans l'enseignement obligatoire, principalement en raison de l'organisation de nos systèmes éducatifs et de la place que peut s'octroyer une nouvelle discipline scolaire (*ibidem*). Cela reste néanmoins à confirmer.

On peut distinguer plusieurs enjeux d'un enseignement généralisé de l'informatique, prenant l'aspect « pensée computationnelle » dans un sens transdisciplinaire et/ou métacognitif.

- L'enjeu scientifique : la nécessité de comprendre le fonctionnement du monde numérique, à l'instar de la nécessité de comprendre le fonctionnement du monde physique.

- L'enjeu sociétal : la nécessité d'être à même d'évaluer de façon informée et fondée les impacts de l'informatique sur la société et de contribuer à façonner son développement.

- L'enjeu transdisciplinaire : l'acquisition des outils à la fois mentaux et techniques au service d'autres sciences ou contribuant à des stratégies plus générales de résolution de problèmes.

Une question se pose avec insistance : quels que soient les choix pour les curricula, quels enseignants seront à même de les mettre en œuvre ? Quelle formation pourrait être proposée à ces enseignants ? Même si la pensée informatique reste encore mal définie, on peut certainement déplorer le manque d'enseignants ayant les dispositions et les compétences pour l'enseigner.

4. De la pensée informatique aux compétences de base

Si l'on se limite, pour simplifier le cadre conceptuel de la pensée informatique, à ce qui concerne la résolution de problèmes de manière effective et efficiente, algorithmiquement, avec ou sans machine, différentes facettes ont été mises en exergue : décomposition, abstraction, conception d'algorithmes, débogage, itération et généralisation (Shute *et al.*, 2017).

Prenons l'exemple du débogage. Shute et ses co-auteurs le définissent comme la détection et l'identification des erreurs, leur correction, quand

un programme ne fonctionne pas comme il devrait. Dans (Bers *et al.*, 2014), il est considéré comme au cœur de la pensée informatique, constituant une mesure du développement de la pensée informatique chez les enfants. Ces auteurs identifient les différentes étapes par lesquels passent les enfants pour déboguer : d'abord reconnaître un écart à l'état final attendu ou que quelque chose ne fonctionne pas comme attendu, puis choisir de maintenir l'objectif initial ou se tourner vers une alternative, ensuite formuler des hypothèses sur la cause du problème et enfin tenter de résoudre le problème. Il s'agit d'un apprentissage lié à l'informatique et plus généralement à l'ingénierie.

Plus simplement, la pensée informatique tourne autour du fait d'être capable de formuler un problème, de le comprendre et de le résoudre. Avec une telle définition, on peut s'interroger sur l'absence de référence à Polya (1965), très populaire au cours des développements de l'intelligence artificielle dans les années 1970 et 1980. Mais, comme on l'a vu, remonter à Papert dans l'histoire de la pensée informatique oriente sur les questions d'enseignement et d'apprentissage, et même sur les apprentissages de base, les apprentissages pour tous. Cela conduit aux questions de littératie, liées aux compétences de base comme la lecture et l'écriture.

C'est une position défendue par diSessa (2000), dans sa vision de la culture informatique universelle. Selon lui, les dispositifs informatisés (ordinateurs) forment la base pour une nouvelle littératie qui va modifier la manière avec laquelle les humains pensent et apprennent, et chacun sera un créateur aussi bien qu'un consommateur de formes expressives dynamiques et interactives. Cette perspective est également défendue par Jacob et Warschauer (2018) et par Wing : « *My grand vision is that computational thinking will be a fundamental skill—just like reading, writing, and arithmetic—used by everyone by the middle of the 21st century* » (Wing, 2017).

Au-delà des traditionnelles littératies informationnelles (*information literacy*) et numériques (*digital literacy*), on peut faire le lien avec la translittératie, qui s'exerce sur une multitude de supports et de médias, étudiée dans le projet ANR Translit (<https://anr.fr/Projet-ANR-12-CULT-0004>). C'est également la manière contemporaine de voir les questions d'organisation, avec des synthèses entre des traditions documentaires et les offres informatiques, conduisant à la science de l'organisation (Glushko, 2016).

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

La notion d'écriture est centrale et les effets de l'écriture sont bien exposés par Jack Goody (1977). L'apprentissage du calcul se fait aussi par la manipulation d'objets et par la lecture et la transformation d'écritures. Ces écritures fournissent une représentation de l'ordre des nombres et des quantités, les opérations s'effectuent par l'intermédiaire des écritures. On a beaucoup écrit sur l'externalisation de la mémoire avec les technologies informatiques, et il est intéressant de revenir au projet MyLifeBits (décrit sur les sites <https://www.microsoft.com/en-us/research/project/mylifebits/> et <https://fr.wikipedia.org/wiki/MyLifeBits>), caractéristique des phénomènes d'externalisation, de conservation et d'organisation de ces mémoires. Les technologies informatiques actuelles n'offrent-elles pas une nouvelle forme d'externalisation de la pensée, en tous cas d'une certaine forme de pensée, les étapes successives d'un raisonnement étant traduites dans des écritures, extension de la logique à d'autres formes de raisonnement ?

Ainsi la question est celle de l'évolution du lire-écrire-computer dans un univers où les dispositifs informatisés et autres objets connectés sont omniprésents, ambiants même. On peut proposer le triptyque lire-écrire-computer (Bruillard, 2012). Si le développement de cette alphabétisation s'inscrit dans un courant d'émancipation humaine, il ne devrait pas conduire à oublier qu'il convient également de contrôler les machines, de limiter leurs moyens d'action. La vague actuelle d'engouement pour l'intelligence artificielle peut être préoccupante, si elle conduit à laisser trop de pouvoir aux machines et aux organisations qui les utilisent, sans développer la littératie liée à la pensée informatique à la hauteur de l'ambition forte de cette éducation.

RÉFÉRENCES

Académie des sciences (mai 2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre* (Rapport). Institut de France-Académie des sciences. https://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf

Arsac, J. (1981). Annexe 1. Dans J. C. Simon (dir.), *L'éducation et l'informatisation de la société. Rapport au Président de la République* (p. 152-165). La Documentation française.

Atwood, J. (2012). Please don't learn to code [article de blog]. <https://blog.codinghorror.com/please-dont-learn-to-code/>.

Baron, G.-L. (1990). Note de synthèse. *Revue française de pédagogie*, 92(1), 57-77.

Barr, V. et Stephenson, C., (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.

Bers, M. U. (2017). The Seymour test: Powerful ideas in early childhood education. *International Journal of Child-Computer Interaction*, 14, 10-14. <https://doi.org/10.1016/j.ijcci.2017.06.004>

Bers, M. U. (2018). La programmation en tant que place de jeu développementale: la pensée informatique et la robotique dans la petite enfance. Communication présentée au colloque *Didapro 7 – DidaSTIC: de 0 à 1 ou l'heure de l'informatique à l'école*. <https://hal.archives-ouvertes.fr/hal-01748783>

Bers, M. U., Flannery, L., Kazakoff, E. R. et Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.

Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). *Developing computational thinking in compulsory education. Implications for policy and practice* (Rapport EUR 28295). Luxembourg: Publications Office of the European Union. <https://doi.org/10.2791/792158>

Bocconi, S., Chiocciariello, A. et Earp, J. (janvier 2018). *The nordic approach to introducing computational thinking and programming in compulsory education*. (Report prepared for the Nordic@BETT2018 Steering Group). <https://doi.org/10.17471/54007>

Boulle, F. (1984). Informatique à l'école. Introduction et éléments d'histoire. *Bulletin de l'EPI*, 6. <https://www.epi.asso.fr/revue/dossiers/d06p005.htm>

Bruillard, É. (2012). Lire-écrire-computer : émanciper les humains, contrôler les machines. Dans e-Dossier de l'Audiovisuel « Éducation aux cultures de l'information », INA. <http://www.epi.asso.fr/revue/articles/a1209d.htm>

Bruillard, É. (2017). Enseignement de l'informatique entre science et usages créatifs : quelle scolarisation ? Dans J. Henry, A. Nguyen et É. Vandepuut (dir.), *L'informatique et le numérique dans la classe. Qui, quoi, comment ?* (p. 205-218). Belgique : Presses Universitaires de Namur.

Chessel Lazzarotto, F. (2018). Former à la programmation en primaire, une form'action : robots d'Evian 2015-2018. Dans G. Parriaux, J. P. Pellet, G. L. Baron, E. Bruillard, V. Komis (dir.), *De 0 à 1 ou l'heure de l'informatique à l'école. Actes du Colloque Didapro 7 - DidaSTIC* (p. 117-128). Bern, Suisse : Peter Lang.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. et Woollard, J. (2015). Computational thinking A guide for teachers. Computing at School. <https://www.computingschool.org.uk/computationalthinking>

Delmas-Rigoutsos, Y. (2018). Proposition de structuration historique des concepts de la pensée informatique fondamentale. Dans G. Parriaux, J. P. Pellet, G. L. Baron, E. Bruillard, V. Komis (dir.), *De 0 à 1 ou l'heure de l'informatique à l'école. Actes du Colloque Didapro 7 - DidaSTIC* (p. 31-60). Bern, Suisse : Peter Lang.

Denley, P. (1994). Models, sources and users: Historical database design in the 1990s. *History and Computing*, 6(1), 33-43.

Denning, P. J. (2003). Great principles of computing, *Communications of the ACM*, 46(11), 15-20. <http://hdl.handle.net/10945/35508>

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. <https://doi.org/10.1145/2998438>

**Béatrice DROT-DELANGE, Jean-Philippe PELLET,
Yannis DELMAS-RIGOUTSOS, Éric BRUILLARD**

Department for Education. (2013). *The National Curriculum in England*. Disponible en ligne à www.education.gov.uk/nationalcurriculum.

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge : MIT Press.

Drot-Delange, B. (2013). Enseigner l'informatique débranchée : analyse didactique d'activités (p. 1-13). Dans *Actes du congrès Actualité de la Recherche en Education et en Formation (AREF 2013)* (p. 1-13). https://halshs.archives-ouvertes.fr/sic_00955208

Drot-Delange, B. et Tort, F. (2018). Résolution de défis et pensée informatique. Présenté aux *10e rencontres scientifiques de l'ARDIST*. <https://hal.archives-ouvertes.fr/hal-01851772/document>

Glushko, R.-J. (dir.). (2016). *The discipline of organizing: Professional edition* (4e éd.). Sebastopol, CA : O'Reilly.

Goody, J. (1977). *The Domestication of the Savage Mind*. Cambridge University Press.

Grover, S. et Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <http://journals.sagepub.com/doi/abs/10.3102/0013189X12463051>

Guzdial, M. (2010). Does contextualized computing education help? *ACM Inroads*, 1(4), 4-6.

Hartmann, W., Näf, M. et Reichert, R. (2012). *Enseigner l'informatique (Traduit de l'allemand)*. Paris : Springer.

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. *IEEE Frontiers in Education Conference (FIE)*. <https://doi.org/10.1109/FIE.2016.7757410>

Howland, K., Good, J., Robertson, J. et Manches, A. (dir.) (2018). Computational thinking and coding in childhood [numéro thématique]. *International Journal of Child-Computer Interaction*.

Hickmott, D., Prieto-Rodriguez, E. et Holmes, K. (2017). A scoping review of studies on computational thinking in K-12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4(1), 48-69. <https://link.springer.com/article/10.1007/s40751-017-0038-8>

Hsu, T. C., Chang, S. C. et Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310. <https://doi.org/10.1016/j.compedu.2018.07.004>

Jacob, S. R. et Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, 1(1). <https://doi.org/10.26716/jcsi.2018.01.1>

Kalelioglu, F., Gülbahar, Y. et Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583-596. <https://pdfs.semanticscholar.org/e4f9/4b4e16a87e4b815893388c63c096038a69b8.pdf>

Kirwan, C., Costello, E. et Donlon, E. (2018). Computational Thinking and Online Learning: A Systematic Literature Review. Dans *Proceedings of the 17th European Conference on eLearning (ECEL 2018)* (p. 650-657). Reading, UK : Academic Conferences and Publishing International Limited.

Knobelsdorf, M. et Tenenberg, J. (2013). The context-based approach IniK in light of situated and constructive learning theories. Dans *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. Proceedings of the 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (ISSEP 2013)* (p. 103-114). Berlin : Springer.

Lockwood, J. et Mooney, A. (2017). Computational Thinking in Secondary Education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*. 2(1).
<http://ijcses.org/index.php/ijcses/article/view/26>

Mazzone, M. (2018). Andy Warhol: Computational thinking, computational process. *Leonardo Music Journal*. https://doi.org/10.1162/LEON_a_01574

Papert, S. (1981). *Jaillissement de l'esprit. Ordinateurs et apprentissage*. Edition 1999. Flammarion. (Ouvrage original publié en 1980 sous le titre *Mindstorms - Children, computers and powerful ideas*. USA : Basic Books).

Papert, S. (2000). What's the big idea? Toward a pedagogy of idea power. *IBM systems journal*, 39(3.4), 720-729.

Parriaux, G., Pellet, J.-P., Baron, G.-L., Bruillard, E. et Komis, V. (dir.). (2018). De 0 à 1 ou l'heure de l'informatique à l'école. Actes du Colloque Didapro 7 - DidaSTIC. Bern, Suisse : Peter Lang.

Polya, G. (1965). *Comment poser et résoudre un problème*. Paris : Dunod. (Ouvrage original publié en 1945 sous le titre *How to solve it: A new aspect of mathematical method*. Princeton University Press).

Rogalski, J. (1987). Acquisition de savoirs et savoir-faire en informatique. *Cahiers de Didactique des Mathématiques*, 43.

Selby, C. et Woollard, J. (2013). *Computational thinking: the developing definition* (Rapport). <https://eprints.soton.ac.uk/356481/>

Shute, V. J., Sun, C. et Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
<https://www.sciencedirect.com/science/article/pii/S1747938X17300350>

Voogt, J., Fisser, P., Good, J., Mishra, P. et Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728.
<https://doi.org/10.1007/s10639-015-9412-6>

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3), 33-35.
<https://doi.org/10.1145/1118178.1118215>.

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14.
<https://doi.org/10.17471/2499-4324/922>



Comités

1. Rédacteur en chef

Sébastien GEORGE • LIUM, Le Mans Université, Laval

2. Comité de rédaction

Georges-Louis BARON • EDA, Université de Paris

Monique BARON • LIP6, Sorbonne Université

Laetitia BOULC'H • EDA, Université de Paris

Éric BRUILLARD • EDA, Université de Paris

Pierre-André CARON • CIREL, Université Lille

Christophe DESPRÈS • LIUM, Le Mans Université

Nour EL MAWAS • CIREL, Université Lille

Sébastien GEORGE • LIUM, Le Mans Université, Laval

Monique GRANDBASTIEN • LORIA, Université de Lorraine

Richard HOTTE • LICEF, Télé-université, Université du Québec,
Montréal, Canada

Pierre JACOBONI • LIUM, Le Mans Université

Élise LAVOUÉ • IAE de Lyon, Université Jean Moulin Lyon 3

Vanda LUENGO • LIP6, Sorbonne Université

Agathe MERCERON • Université de Berlin, Allemagne

Gaëlle MOLINARI • TECFA, Unidistance, Genève, Suisse

Chrysta PÉLISSIER • Praxiling, Université Montpellier 3

Jean-Luc RINAUDO • Civiic, Université de Rouen

3. Comité de parrainage scientifique

Nicolas BALACHEFF • Laboratoire d'Informatique de Grenoble, CNRS

Stefano CERRI • LIRMM & Université de Montpellier 2

Christian DEPOVER • Université de Mons, Belgique

Alain DERYCKE • TRIGONE, Université de Lille

Pierre DILLENBOURG • École polytechnique fédérale de Lausanne, Suisse

Claude FRASSON • Université de Montréal, Canada

Catherine GARBAY • CNRS, laboratoire d'Informatique de Grenoble

Gilles GAUTHIER • UQAM, Canada

Guy GOUARDÈRES • ISIHM, Université de Pau
Ulrich HOPPE • Université de Duisbourg, Allemagne
Jean-Marc LABAT • LIP6, Sorbonne Université
Patrick MENDELSON • LSE, IUFM de Grenoble
Jean-François NICAUD • LIG, Université Grenoble Alpes
Gilbert PAQUETTE • LICEF, Télé-université, Université du Québec,
Montréal, Canada
Jacques PERRIAULT • Université Paris Nanterre
Janine ROGALSKI • Laboratoire « Cognition et activités finalisées »,
Université de Vincennes-Saint-Denis
Maria Felisa VERDEJO • Universidad nacional de educación a distancia,
Espagne

4. Comité de lecture

Michel ARNAUD • Université Paris Nanterre
François-Xavier BERNARD • EDA, Université de Paris
Mireille BÉTRANCOURT • TECFA, Université de Genève, Suisse
Jacques BÉZIAT • CIRNEF, Université de Caen Normandie
Bernard BLANDIN • CREF, Université Paris Nanterre et CESI
François BOUCHET • LIP6, Sorbonne Université
Julien BROISIN • IRT, Université de Toulouse Paul Sabatier
Thibault CARRON • LIP6, Sorbonne Université et Université de Savoie
Mont-Blanc
Ulrich CARSTEN • EdTec Lab, DFKI GmbH, Sarrebrück, Allemagne
Thierry CHANIER • LRL, Université Clermont Auvergne
Ghislaine CHARTRON • CNAM, Paris
Christophe CHOQUET • LIUM, Le Mans Université, Laval
Philippe COTTIER • CREN, Université de Nantes
Jacques CRINON • INSPÉ, Université Paris Est Créteil
Bruno DE LIÈVRE • Université de Mons, Belgique
Nicolas DELESTRE • LITIS, INSA de Rouen
Élisabeth DELOZANNE • LIP6, Sorbonne Université
Michel DESMARAIS • École polytechnique de Montréal, Canada
Cyrille DESMOULINS • LIG, Université Grenoble Alpes
Philippe DESSUS • LSE, Université Grenoble Alpes
Angélique DIMITRACOPOULOU • LTEE, Université d'Egée, Grèce
Béatrice DROT-DELANGE • ACTé, Université Clermont Auvergne
Aude DUFRESNE • ESI, Université de Montréal, Canada
Cédric FLUCKIGER • Théodile-CIREL, Université Lille
Serge GARLATTI • Lab-STICC, IMT Atlantique, Brest

Jean-Marie GILLIOT • Lab-STICC, IMT Atlantique, Brest
Brigitte GRUGEON • LDAR, INSPÉ, Université Paris Est Créteil
Viviane GUÉRAUD • LIG, Université Grenoble Alpes
Nicolas GUICHON • ICAR, Université Lumière Lyon 2
Nathalie GUIN • LIRIS, Université Lyon 1
France HENRI • LICEF, Télé-université, Université du Québec,
Montréal, Canada
Pierre JARRAUD • FOAD, Sorbonne Université
Michelle JOAB • LIRMM, Université Montpellier 2
Céline JOIRON • MIS, Université de Picardie Jules Verne, Amiens
Mehdi KHANEBOUBI • STEF, ENS Paris-Saclay
Vassilis KOMIS • Université de Patras, Grèce
Thérèse LAFERRIÈRE • TACT, Université Laval, Canada
Françoise LE CALVEZ • LIP 6, Sorbonne Université
Marie LEFÈVRE • LIRIS, Université Lyon 1
Dominique LENNE • Heudiasyc, Université de Technologie de
Compiègne
Pascal LEROUX • CREN, Le Mans Université
Paul LIBBRECHT • Leibniz Institute for Research and Information in
Education, Allemagne
Cabral LIMA • Université Fédéral de Rio de Janeiro, Brésil
Domitile LOURDEAUX • Heudiasyc, Université de Technologie de
Compiègne
Pascal MARQUET • LISEC, Université de Strasbourg
Jean-Charles MARTY • LIRIS, Université de Savoie
André MAYERS • Université de Sherbrooke, Canada
Christine MICHEL • TECHNÉ, Université de Poitiers
Roger NKAMBOU • GDAC, Université du Québec à Montréal, Canada
Thierry NODENOT • LIUPPA, Université de Pau et des Pays de l'Adour,
Bayonne
Daniel PERAYA • TECFA, Université de Genève, Suisse
Yvan PETER • LIFL, Université Lille, Villeneuve d'Ascq
Julia PILET • LDAR, INSPÉ, Université Paris Est Créteil Val de Marne
Dominique PY • LIUM, Le Mans Université
Christophe REFFAY • ELLIAD, INSPÉ, Université de Franche-Comté
Éric SANCHEZ • CERF, Université de Fribourg, Suisse
Karim SEHABA • LIRIS, Université Lyon 2
Nicolas SZILAS • TECFA, Université de Genève, Suisse
Pierre TCHOUNIKINE • LIG, Université de Grenoble

André TRICOT • CERFI, INSPÉ de Midi-Pyrénées & Université Le-Mirail-
Toulouse 2

Nicolas VAN LABEKE • Learning Sciences Research Institute,
University of Nottingham, UK

Jean VANDERDONCKT • ISYS, Université catholique de Louvain,
Belgique

Kalina YACEF • Université de Sydney, Australie

En mémoire d'anciens membres de comités :

Erik DUVAL • Université de Louvain, Belgique

Jacques PERRIAULT • Université de Paris Nanterre

François VILLEMONTÉIX • CIREL, Université de Lille

**Nous remercions les personnes extérieures aux comités
qui ont relu pour ce numéro :**

Brigitte DENIS • CRIFA, Liège Université

Luc MASSOU • CREM, Université de Lorraine

Margarida ROMERO • INSPÉ, Université Côte d'Azur

Amel YESSAD • LIP6, Sorbonne Université



ISBN 978-2-901384-02-1
DOI:10.23709/sticef.26.1