



# Enseigner, apprendre, former à l'informatique à l'école : regards croisés

Édition coordonnée par Cédric Fluckiger, Laetitia Boul'ch,  
Sandra Nogry et Christophe Reffay



# **Enseigner, apprendre, former à l'informatique à l'école : regards croisés**

Illustration de couverture

Crédit : Julia Fraud, <https://juliafraudillumed.ultra-book.com/>



# Enseigner, apprendre, former à l'informatique à l'école : regards croisés

Édition coordonnée par Cédric FLUCKIGER, Laetitia BOULC'H,  
Sandra NOGRY et Christophe REFFAY

Université Paris Cité

2024

Édition réalisée avec le soutien de l'Agence nationale de la recherche (financement du projet IE CARE « Informatique à l'école : conceptualisations, accompagnement, ressources »), de l'université de Lille et de l'ULR 4354 - CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, F-59000 Lille







<https://doi.org/10.53480/2024iecare/>

ISBN 978-2-7442-0212-4 (PDF)  
ISBN 978-2-7442-0211-7 (imprimé)

### **Licence Creative Commons**

© Livres publiés en accès ouvert selon les termes de la licence Creative Commons Attribution License 4.0 (CC BY), qui permet l'utilisation, la distribution et la reproduction sans restriction et sur tout support, à condition que l'œuvre originale soit correctement citée : <https://creativecommons.org/licenses/by/4.0/>

La licence CC BY s'applique à l'ensemble de l'ouvrage sauf mentions contraires.

Les images reproduites avec l'autorisation d'un tiers, sont identifiées par la mention d'un crédit ou *copyright* dans leur légende. Il vous appartient, si vous souhaitez reproduire à votre tour ces images, d'obtenir l'autorisation des ayants droit.

© Les autrices et auteurs, 2024

# Préface

L'informatique est un fait social multiforme, maintenant largement intégré dans la société française de ce début de XXI<sup>e</sup> siècle. Ses diverses manifestations étonnent, suscitent une adoption souvent teintée de crainte, parfois le rejet, rarement l'indifférence. Pourtant, c'est un fait éducatif relativement peu pris en compte à l'école primaire.

Au début des années 1990, des ordinateurs dotés de fonctions de manipulation directe (en général via une souris) et de bonnes capacités graphiques ainsi que de supports de mémoire adaptés (des CD-ROMs à l'époque) sont devenus accessibles au grand public. Les autorités pédagogiques ont alors considéré qu'il n'y avait pas lieu d'organiser un enseignement d'informatique dans la formation générale. Pour elles, l'outil informatique, convivial, ne nécessitait qu'un peu de familiarisation, peut-être l'acquisition par la pratique de quelques compétences (en particulier, en technologie collège) mais en aucun cas un curriculum spécifique.

Cette vision naïve a fini par s'estomper. Depuis le début des années 2000, il a été progressivement accepté qu'il y avait des compétences à acquérir, puis des connaissances. La question devenait alors de préciser lesquelles étaient indispensables et de concevoir un curriculum les articulant. Cela a occupé chercheurs, militants (on ne parlait pas encore d'influenceurs) et décideurs pendant pas mal de temps.

Dans la décennie 2010, des enseignements d'informatique ont été introduits dans le second degré et, en 2020, un certificat d'aptitude au professorat de l'enseignement du second degré (CAPES), concours de recrutement de professeurs d'informatique, a finalement été institué, ce qui marque de manière officielle la naissance de l'informatique comme discipline scolaire. Fait significatif d'un rapport de forces favorable, un concours d'agrégation a été créé, dès 2022, donnant finalement suite à une des propositions du rapport Simon en 1980 au président de la République.

Il n'en a pas été ainsi dans le premier degré. Des activités informatiques ont été organisées à l'école depuis les années 1980, mais elles sont restées occasionnelles, souvent menées à l'occasion d'innovations conduites par des personnes très motivées au service de l'apprentissage de leurs élèves. Parmi ces activités, celles qui sont liées aux algorithmes et à la programmation (à ce qu'on appelle souvent, de manière sans doute abusive, la « pensée informatique ») ont une place à part. Mais elles n'ont pour l'instant pas trouvé de place comme matière d'enseignement et n'ont pas fait l'objet d'une grande priorité.

Des recherches se sont intéressées dès la fin des années 1970 à l'apprentissage de la programmation par de jeunes enfants, en particulier autour du langage Logo, qui a fourni des situations conduisant à programmer des robots tangibles (la « tortue Logo ») ou à organiser des interactions entre entités simulées sur écran (« tortue-écran »). Dans cette lignée, une innovation majeure sera apportée par le système Scratch et son adaptation pour de jeunes enfants, ScratchJr.

Ce flux de recherches a permis de produire un ensemble de résultats mais aussi de nouvelles problématisations quant à la manière de considérer une initiation à l'informatique à l'école primaire.

Le projet pluridisciplinaire « Informatique à l'école : conceptualisations, accompagnement, ressources » IE CARE (associant des chercheurs en sciences humaines et sociales et en informatique), accepté par l'Agence nationale de la recherche en 2018, s'inscrit dans cette filiation.

Son objectif principal a été de comprendre et de proposer des conditions et des modalités durables pour un enseignement de l'informatique à l'école obligatoire. Trois axes de développement reliés par une tâche transversale ont été définis : le premier axe a visé à délimiter un ensemble de contenus informatiques enseignables, le deuxième à concevoir et mettre à l'épreuve des scénarios pédagogiques et des ressources pour soutenir les pratiques d'enseignement et d'apprentissage de l'informatique à l'école et au collège. Enfin, le troisième axe s'est orienté vers la mise en place d'un cadre d'accompagnement pour les enseignants et les formateurs en informatique.

Le déroulement du projet a malheureusement été fortement perturbé par plusieurs causes. D'abord, François Villemonteix qui était à son initiative est décédé subitement en 2018. Il avait commencé sa carrière comme enseignant du primaire et en avait une connaissance très approfondie puisqu'il a été ensuite inspecteur de l'Éducation nationale. Il s'était très tôt intéressé à l'informatique en éducation et avait suivi un parcours d'études universitaires le menant jusqu'à devenir professeur de sciences de l'éducation à l'université de Lille. Il avait notamment joué un rôle important dans le projet Didactique et apprentissage de l'informatique à l'école (DALIE), qui a précédé IE CARE. Sa disparition a été durement ressentie. Le relais a cependant été rapidement pris par Cédric Fluckiger, avec l'aide des responsables d'axes.

Par ailleurs, à partir de 2020, la crise sanitaire du coronavirus a considérablement entravé les travaux : les nouvelles nécessités de « continuité pédagogique » et leurs exigences d'enseignement à distance sont brutalement venues au premier plan. La coordination entre des équipes de recherche se connaissant déjà n'en a pas trop pâti, mais les enseignants ont souffert d'un surcroît de travail et de stress. Beaucoup d'observations en classe ont dû être reportées à des jours meilleurs et la co-conception de ressources en relation avec des praticiens a pris un retard qui n'a pu être pleinement rattrapé.

Pour autant, la recherche menée a permis une production scientifique conséquente, dont ce livre présente une synthèse. Ses coordinateurs ont choisi un plan qui ne reprend pas les trois axes initiaux, mais s'organise, après un chapitre de contextualisation historique, autour de trois grandes parties : enseigner ; apprendre ; former. Ce choix est logique car ce sont là les grands aspects fondamentaux de tout enseignement. On y trouvera des analyses d'instructions officielles, de pratiques d'enseignants, d'activités d'élèves, d'actions de formation et des réflexions sur la scénarisation de ressources destinées à accompagner l'action enseignante.

L'ensemble me semble bien illustrer la multiplicité des questions que pose le fait informatique à l'école primaire. L'enjeu n'y est pas seulement l'acquisition d'une « pensée informatique », mais il enveloppe ou intersecte de nombreux champs, dont le moindre n'est pas celui des conditions permettant aux enseignants de s'approprier des ressources, de les modifier pour rendre viable leur enseignement : ressources matérielles et humaines, ressources immatérielles d'accompagnement. . .

Bien entendu, de nombreuses questions subsistent relativement aux problèmes qui se posent pratiquement pour faire exister une éducation à l'informatique à ce niveau scolaire. Comment les étudier ? Comme l'indique Eric Bruillard dans la postface, l'enjeu est de prendre en compte, d'une manière qu'on pourrait qualifier de systémique, à la fois les contenus, les ressources, et les situations éducatives.

Fruit d'une approche pluridisciplinaire, cet ouvrage donne une série de clés permettant de comprendre un mouvement qui est loin d'être achevé et dont l'importance pour les jeunes générations ne peut être sous-estimée.

Georges-Louis BARON

# Introduction

Cédric FLUCKIGER<sup>1</sup>  
Laetitia BOULC'H<sup>2</sup>  
Sandra NOGRY<sup>3</sup>  
Christophe REFFAY<sup>4</sup>

1. Université de Lille, ULR 4354, CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, 59000 Lille, France
2. Université Paris Cité, Laboratoire EDA, 75006 Paris, France
3. Cergy Paris Université, INSPE de Versailles, Laboratoire EMA, 95000 Cergy-Pontoise, France
4. Université de Franche-Comté, ELLIADD, 25000 Besançon, France

## Le projet IE CARE

Le projet « Informatique à l'école : conceptualisations, accompagnement, ressources » IE CARE, dont est tiré cet ouvrage, visait à comprendre et proposer des conditions et modalités durables d'un enseignement de l'informatique à l'école obligatoire. Il s'est déroulé de 2019 à 2023, mobilisant une communauté de chercheurs en sciences de l'éducation et de la formation, informatique, psychologie, sciences du langage notamment.

Ce projet faisait suite à un autre projet (projet Didactique et apprentissage de l'informatique à l'école – DALIE), financé par l'Agence nationale de la recherche (ANR), qui s'était intéressé à la mise en œuvre d'une approche curriculaire de l'enseignement de l'informatique à l'école primaire, notamment à l'occasion de séances en rapport avec la robotique pédagogique, le traitement de données numérisées et la familiarisation avec les enjeux liés au développement massif de ces technologies dans nos sociétés.

Concernant les terrains de recherche, deux objectifs majeurs ont été poursuivis dans le projet IE CARE :

- étudier les pratiques et représentations existantes, c'est-à-dire celles qui existent avant même nos actions de formation et hors des scénarios et outils que nous cherchons à développer ;
- concevoir et développer des outils, à travers des processus collaboratifs avec des formateurs et enseignants.

Par ses objectifs, ses cadres théoriques et sa méthodologie générale, ce projet relève de la *didactique de l'informatique*. Ce qui caractérise la didactique et la spécifie par rapport aux autres disciplines

qui étudient les phénomènes d'enseignement et d'apprentissage, c'est qu'elle se focalise sur les contenus enseignés et appris, en cherchant à investiguer les processus, difficultés, conditions qui sont propres à un type de contenu, une matière scolaire, une discipline. Comme le formule Johsua dans sa discussion avec Lahire :

Il est possible que ce soit pareil d'enseigner la poésie ou les mathématiques, mais avant de conclure dans ce sens, il faut d'abord se demander ce qui est spécifique aux mathématiques, ce qui est spécifique à la poésie. (Lahire et Johsua, 1999, p. 31)

Concernant l'informatique, la didactique peut donc étudier finement la mise en place des enseignements d'informatique (qui est décrite dans le premier chapitre de l'ouvrage), en interrogeant ce qu'il y a de spécifique à ces contenus :

- en lien avec la nature des savoirs en jeu : apprendre à faire une boucle dans un environnement de programmation pour enfants est une tâche cognitive spécifique, différente, par exemple, de l'apprentissage d'une poésie et qui posera des difficultés et des obstacles didactiques particuliers ;
- en lien avec l'organisation curriculaire des contenus : il n'est pas neutre par exemple qu'au collège, l'enseignement de l'informatique ne relève pas d'une matière spécifique mais soit pris en charge par les mathématiques et la technologie ;
- en lien avec la reconnaissance sociale qu'ont ces savoirs au sein de la société : considère-t-on qu'il y a besoin de les apprendre ou que le seul usage par les enfants d'outils numériques suffit à en faire des « natifs numériques » compétents ? Considère-t-on qu'il est nécessaire de comprendre ce qui se passe derrière l'écran ou que des compétences d'usages peuvent suffire ? Considère-t-on que l'informatique est plutôt une affaire de garçons ou qu'il faut être un « geek » pour y exceller (un peu comme l'idée d'une « bosse des mathématiques » a pu servir à bien des apprenants à dire « les maths, ce n'est pas pour moi ») ?

Pour répondre à ces questions proprement didactiques, il est nécessaire de bien comprendre la nature du savoir enseigné. C'est pour cette raison que, dans une démarche didactique, doivent collaborer à la fois des spécialistes de la discipline de référence (l'informatique telle qu'elle s'enseigne dans le supérieur ou se pratique dans les travaux de recherche en informatique) et des spécialistes de l'enseignement scolaire, habitués aux méthodes de recherches et concepts permettant de décrire ces enseignements.

Toutefois, si l'ouvrage relève de la didactique (de l'informatique), tous les auteurs ne s'inscrivent pas dans les cadres théoriques

de la didactique. En effet, pour éclairer les phénomènes d'enseignement, d'apprentissage et la formation à cet enseignement, le projet a fait appel à des spécialistes de divers points de vue sur l'enseignement et l'apprentissage : didacticiens, psychologues, linguistes. . . Ainsi, certains concepts spécifiquement didactiques sont mobilisés (les notions de disciplines, de classe de situations, de champ conceptuel, de « rapport à »..), alors que d'autres travaux mobilisent d'autres concepts, construits dans d'autres disciplines.

Par ce croisement des regards, cet ouvrage rend compte de travaux de recherche mais se veut à destination d'un large lectorat : enseignants-chercheurs, étudiants en master d'enseignement, enseignants, conseillers pédagogiques, concepteurs de ressources pour la classe, éditeurs de manuels scolaires, etc. Il porte donc une exigence d'écriture scientifique, qui veut que les affirmations soient étayées (par nos propres résultats empiriques ou par une revue de littérature de recherche), que tout ce qui peut faire résultat soit soumis à une entreprise de justification collective, qui est la marque du discours scientifique. Les chapitres de cet ouvrage sont des chapitres problématisés, les questions initiales sont envisagées par le prisme de cadres théoriques qui sont explicités. Dans le même temps, l'ouvrage cherche à éviter les formes de discours abscons et à destination d'un lectorat d'initiés. Il vise à une forme de diffusion des savoirs, sans rien sacrifier à la rigueur du discours.

En ce sens, l'ouvrage s'inscrit dans le projet même des recherches en didactique, qui sont des recherches impliquées. Le projet de construction de connaissances sur les dynamiques, processus et difficultés d'enseignement et d'apprentissage n'a de sens que s'il rencontre les savoirs issus de la pratique, que portent les praticiens, enseignants, conseillers, etc. La didactique n'est en effet pas normative, au sens où il ne s'agit pas de proposer des recettes toutes faites pour bien enseigner. On sait depuis longtemps qu'une telle démarche ne donne pas de résultats, les conditions et pratiques effectives étant souvent bien éloignées des laboratoires. En revanche, les résultats s'inscrivent dans ce que Reuter nommait un « horizon praxéologique ». Le lecteur ne trouvera donc pas ici l'illusion d'une « bonne méthode » à appliquer ou de l'outil parfait pour surmonter les difficultés des élèves. Il trouvera en revanche des éléments de compréhension, des points d'attention qui, associés aux savoirs pratiques des enseignants, permettront à ceux-ci de mieux saisir les conditions et les enjeux d'enseigner l'informatique dès le plus jeune âge.

Pour comprendre les conditions d'un enseignement de l'informatique, l'ouvrage rassemble onze contributions qui s'intéressent aux trois processus fondamentaux :

- enseigner (de la part des enseignants);

- apprendre (pour les élèves) ;
- former (les enseignants).

Comprendre l'enseignement, l'apprentissage et la formation nous a conduit à investiguer :

- les prescriptions ;
- les représentations des sujets (élèves et enseignants principalement) sur l'informatique, son importance, son utilité, etc. ;
- les pratiques de classe, celles qui existent dans les classes ou celles que le chercheur peut susciter ;
- les ressources à disposition des élèves et des enseignants pour accompagner ou susciter ces apprentissages.

Il s'est en réalité avéré bien difficile d'enquêter sur les pratiques en classe. Il y a des raisons circonstancielles à ces difficultés : les deux années de confinement et de limitation des cours en présentiel ont rendu l'accès aux classes compliqué, que ce soit pour enquêter sur des pratiques ordinaires ou pour monter des expérimentations. Pour autant, cela révèle aussi qu'au-delà des prescriptions, cet enseignement est relativement peu mis en œuvre en classe.

## Partie 1. Enseigner l'informatique à l'école

La première partie de l'ouvrage, composée de quatre chapitres, porte sur l'enseignement de l'informatique à l'école primaire. Un premier constat y est posé : les prescriptions et recommandations qui portent sur cet enseignement ainsi que les ressources à disposition sont nombreuses ; néanmoins, ces documents sont porteurs de visions de l'informatique et de prescriptions qui peuvent paraître contradictoires, ce qui soulève de nombreuses questions quant à leur mise en œuvre.

Le premier chapitre de l'ouvrage, « Quel enseignement de l'informatique à l'école primaire en France ? Réflexions sur 40 ans de développements », retrace l'histoire de l'enseignement à l'école primaire depuis les années 1980. Georges-Louis Baron revient sur l'évolution des politiques publiques durant ces quatre décennies en précisant la place accordée à la programmation et aux objets d'enseignement qui relèvent de la science informatique au cours de cette période. Ce chapitre revient ensuite sur l'évolution des recherches sur cette question, en mettant en exergue la structuration progressive d'une communauté francophone de recherche sur la didactique de l'informatique. Le chapitre se conclut par la présentation de quelques-uns des enjeux actuels pour ce champ de recherche.

Le deuxième chapitre, « Quel enseignement de l'informatique dans la scolarité obligatoire en France ? Analyse des programmes et manuels », vise à appréhender l'informatique en tant que discipline scolaire en analysant les textes institutionnels, programmes

scolaires et manuels publiés ces dernières années dans le but d'orienter les pratiques des enseignants. Dans ce chapitre, la vision de l'informatique portée par les textes officiels est interrogée, les contenus introduits dans les programmes et les manuels sont analysés, les finalités assignées à ces contenus sont discutées. Au fil de ces analyses, apparaissent des tensions entre une vision de l'enseignement de l'informatique centrée sur l'éducation à la citoyenneté – il s'agit de donner aux enfants des clés pour leur permettre de comprendre le monde numérique qui les entoure – et la volonté de constituer une véritable discipline informatique initiant les élèves à la programmation et à la pensée informatique, nécessaire pour comprendre les évolutions technologiques actuelles telles que l'intelligence artificielle (IA). Des écarts entre les visions portées par les textes officiels, les programmes scolaires et les manuels sont également soulignés. À l'issue de ces analyses, les auteurs ré-interrogent les conditions nécessaires au développement d'une discipline informatique à l'école primaire.

Comment les enseignants se saisissent-ils de ces prescriptions et des ressources à leur disposition pour mettre en place un enseignement de l'informatique dans leur classe? Telle est la question traitée dans les chapitres 3 et 4.

Le chapitre 3 « Pratiques envisagées des enseignants pour un enseignement de l'informatique à l'école primaire », présente les résultats d'enquêtes réalisées sur les pratiques d'enseignement de l'informatique à l'école primaire. Partant du constat que les prescriptions formulées dans les programmes ne sont pas systématiquement mises en œuvre, les auteurs choisissent d'interroger la perception qu'en ont les professeurs des écoles (PE). Il apparaît que ceux-ci ont conscience des enjeux relatifs à cet enseignement, mais ont une représentation floue de ce qui leur est demandé. Leur rapport personnel à l'informatique semble un facteur déterminant pour la mise en œuvre d'un enseignement de contenus relatifs à l'informatique dans leur classe. En articulant des approches issues de la didactique et de l'ergonomie, une analyse des pratiques déclarées est ensuite réalisée; les critères de choix des artefacts utilisés pour ces enseignements font l'objet d'une attention particulière.

Dans le prolongement de cette étude, le chapitre 4 « Quelles ressources pour enseigner l'informatique dans le premier degré? Une étude de cas », propose de mettre à jour l'évolution des ressources utilisées et produites par une professeure des écoles pour enseigner l'informatique dans une perspective diachronique. À partir d'observations de séquences d'initiation à la programmation proposées par cette enseignante durant plusieurs années, les auteurs caractérisent les ressources qu'elle met à la disposition de ses élèves, et leurs évolutions d'une année sur l'autre. Les premières ressources proposées, inspirées d'activités décrites dans son manuel de mathématiques, sont progressivement



adaptées, modifiées, enrichies par d'autres ressources intermédiaires, conçues pour répondre aux besoins des élèves et leur permettre de dépasser les difficultés rencontrées. Comme le soulignent les auteurs, ces évolutions témoignent d'un éloignement d'un ancrage relevant de la didactique des mathématiques, très présent la première année, pour aller vers l'intégration progressive de nouveaux repères et connaissances qui relèvent davantage de la didactique de l'informatique.

## Partie 2. Apprendre l'informatique à l'école

Les trois chapitres qui composent la deuxième partie de cet ouvrage s'intéressent aux apprenants ; d'une part, à la manière dont évoluent leurs rapports à l'informatique et à son apprentissage ; et d'autre part à la manière dont ils s'approprient les objets et les notions d'informatique au sein des situations didactiques qui leurs sont proposées, notamment la programmation de robots, qu'ils soient physiques ou virtuels. Les chapitres apportent également un éclairage sur la manière dont les méthodologies de recherche déployées en sciences de l'éducation, en psychologie et/ou en informatique peuvent apporter des éléments de réponses à ces différentes questions.

Le chapitre 5 « Ruptures et continuités dans les représentations de l'informatique et de son apprentissage chez les élèves de cycles 3 et 4 », s'intéresse à la manière dont les représentations de l'informatique et de son apprentissage se construisent et se transforment chez les élèves, au cours de leur parcours personnel, en référence à leur culture enfantine. Plusieurs constats émanent de l'analyse des entretiens et des enquêtes par questionnaire menés chez des élèves de cycles 3 et 4. D'une part, que leurs représentations s'ancrent dans une culture numérique influencée par leurs pratiques ordinaires de l'informatique, l'école ne jouant qu'un rôle mineur dans la (trans)formation de leurs rapports à l'informatique. D'autre part, que la majorité des élèves développent un rapport pratique à l'apprentissage et adoptent difficilement une posture de secondarisation. L'absence d'une formation spécifique des enseignants (voir chapitres 1 et 3) induit des pratiques d'enseignement où l'informatique joue la fonction d'outil au service d'autres disciplines et est plus rarement considérée comme objet d'apprentissage. Enfin, les auteures notent, chez les plus âgés, une prise de conscience de la part qu'occupe l'informatique dans le monde professionnel, tout en considérant que leurs pratiques personnelles sont davantage source de développement de compétences que ce qu'ils font en informatique au collège.

1.

<https://www.education.gouv.fr/au-bo-special-du-26-novembre-2015-programmes-d-enseignement-de-l-ecole-elementaire-et-du-college-3737>

Face aux prescriptions (2015<sup>1</sup>) visant à démocratiser l'apprentissage de l'informatique, des environnements tels que Scratch et les robots programmables ont (re)fait leur apparition dans les salles

de classe. Les deux chapitres suivants se focalisent sur ces environnements qui sont censés permettre une immersion graduelle dans les concepts fondamentaux de l'informatique en minimisant les barrières traditionnelles liées à la complexité syntaxique.

Le chapitre 6 « Usages des robots programmables BeeBot en classe : des objets pour apprendre, des objets à apprendre », concerne les robots programmables BeeBot, fréquemment utilisés par les enseignants du primaire, dans l'objectif d'apprendre aux élèves à programmer, ou plus prosaïquement, dans l'objectif dans leur apprendre à les utiliser. À partir de l'analyse de l'activité instrumentée des élèves, les auteurs montrent que le recours à ces outils n'est pas sans poser problème aux plus jeunes : difficulté à comprendre le caractère différé de l'exécution du programme ou le principe du stockage des instructions dans une mémoire. . . Face à ces obstacles les enseignants mettent en œuvre des médiations langagières et instrumentales. Si la plupart des élèves arrivent à comprendre le fonctionnement mécanique du robot et à programmer séquentiellement ses déplacements, les auteurs s'interrogent néanmoins sur la compréhension qu'ils ont des notions informatiques sous-jacentes et pointent l'importance, pour l'enseignant, d'identifier les concepts en jeu pour parvenir à proposer des situations adaptées. Les auteurs de cette contribution soulèvent finalement la question du choix des artefacts : les fonctionnalités réduites et l'absence d'interface du robot BeeBot, ne permettant pas de construire des séquences complexes ou de travailler d'autres concepts informatiques (boucle, condition. . .).

Le chapitre 7 « Zoom sur quelques erreurs récurrentes lors des premiers apprentissages en algorithmique », apporte un regard complémentaire sur la compréhension des concepts informatiques par les élèves de primaire et de collège programmant un robot virtuel à partir, cette fois-ci, d'un environnement de programmation avec un langage par blocs (Scratch). Sur la base d'un corpus de plus de 200 000 élèves âgés de 7 à 15 ans, l'auteur analyse les traces d'interaction tout en prenant appui sur la théorie des champs conceptuels de Vergnaud. Cette méthodologie originale mêlant traitements statistiques de nombreuses données quantitatives et finesse des observations, permet d'identifier précisément les stratégies et erreurs récurrentes des élèves. Utilisée ici sur un exemple concret de programmation par blocs de puzzles de programmation mobilisant la notion de boucle, cette méthodologie ouvre de nombreuses perspectives de recherche. Elle permettrait de mieux comprendre comment s'opère l'apprentissage des notions de base de l'algorithmique dans des environnements de programmation par blocs et, dans une plus large mesure, de donner quelques repères didactiques aux enseignants de primaire et de collège chargés de cette initiation.

### Partie 3. Former les enseignants

La troisième et dernière partie de cet ouvrage regroupe quatre chapitres dont les recherches s'appuient sur le recueil de données des acteurs de la formation initiale ou continue des enseignants ainsi que sur la proposition d'un outil d'aide à la conception d'activités dans le champ de la pensée informatique.

Le chapitre 8 « Enseigner l'informatique à l'école primaire : quelques caractéristiques des représentations des formateurs d'enseignants du premier degré », est une enquête auprès des formateurs en charge de la formation continue et de l'accompagnement des professeurs des écoles pour développer leurs compétences en informatique : les enseignants référents pour les usages du numérique et des conseillers pédagogiques du premier degré dans deux académies. Cette enquête constitue un relevé des représentations que se font ces acteurs de l'informatique à l'école. Elle montre en particulier que ces deux types d'acteurs s'accordent à dire que l'informatique (en tant que discipline) n'est pas vraiment inscrite dans les programmes de l'école primaire. Leurs actions de formation portent plus souvent sur les habiletés à enseigner avec les outils informatiques (web radio, tableau numérique interactif – TNI, *padlet*, etc.) qu'à enseigner les fondements de l'informatique en tant que discipline (programmation, robotique, codage de l'information, etc.).

Le chapitre 9 « Connaissances pour enseigner l'informatique : analyse textuelle de productions d'enseignants de l'école primaire », porte plus spécifiquement sur une l'analyse textuelle de ressources produites par les enseignants eux-mêmes pour décrire des séquences d'enseignement de l'informatique qu'ils ont conçues et parfois mises en œuvre dans le cadre de leur formation initiale. La méthode proposée est basée sur des techniques lexicales et statistiques et vise l'analyse de corpus de grande taille en vue de repérer les thèmes récurrents dans un tel corpus. Dans ce chapitre, les auteurs vérifient la cohérence des résultats obtenus par rapport à des analyses antérieures. Ces analyses visent à mettre en évidence les connaissances mobilisées par les enseignants stagiaires. Il apparaît que dans ces productions, la nature des connaissances et du lexique utilisé varie en fonction de l'échelle à laquelle est décrite la séquence d'apprentissage. Ainsi, les notions informatiques sont davantage citées dans les descriptions de la séquence d'apprentissage dans sa globalité en référence aux textes de cadrage que dans la partie des ressources décrivant concrètement les activités à mener en classe avec les élèves. La référence explicite à différentes notions informatiques varie également selon la nature de l'activité proposée (robotique / informatique débranchée / programmation visuelle) et le niveau scolaire des élèves. Ces constats soulèvent différentes questions

relatives à la nature des transpositions didactiques opérées par les enseignants stagiaires lors de la conception de ces séquences.

Le chapitre 10 « Former les enseignants à et par la programmation informatique cocréative en formation initiale » est situé dans le contexte de la formation initiale des enseignants et porte justement sur la programmation. Il décrit un écosystème de recherche création pédagogique dans lequel les étudiants (enseignants stagiaires) développent des compétences en acte pour engager leurs élèves dans la pensée informatique. Guidé par une approche expérientielle, l'environnement proposé par le formateur est peuplé de défis et de ressources, mais surtout d'espaces d'interaction entre pairs permettant la collaboration et la créativité pour toucher à la pensée complexe à travers ces interactions variées. D'un point de vue méthodologique, les chercheurs s'appuient sur les portefeuilles de projets des étudiants pour y mesurer les niveaux de compétence de la pensée complexe, appliqués à la programmation cocréative.

Dans le quatrième et dernier chapitre de cette partie (chapitre 11, « Assister les enseignants du primaire dans l'enseignement de la pensée informatique. Une approche basée sur la scénarisation et vers les *teaching analytics* »), les auteurs présentent la genèse d'un outil de scénarisation d'activités pédagogiques visant l'apprentissage de la pensée informatique. Ce processus, centré sur les utilisateurs, a débuté par une construction négociée d'un modèle de scénarisation d'activités d'enseignement. Ce modèle a ensuite été réifié dans une plateforme accessible en ligne pour soutenir les nombreux enseignants qui débutent dans l'enseignement de la pensée informatique, dans la construction de leurs compétences dans ce domaine. Un exemple permet d'illustrer les fonctionnalités que propose la plateforme. Cette dernière a été testée à plusieurs reprises et les premières traces d'utilisation laissées par les enseignants dans la plateforme sont analysées pour pouvoir envisager de fournir des rétroactions aux utilisateurs lors de la préparation de leurs séances portant sur la pensée informatique. L'ambition de cette plateforme est de proposer un moyen concret et pratique aux enseignants pour soutenir leur formation dans ce domaine qui leur est souvent éloigné.

Pour citer ce chapitre :

Fluckiger, Cédric, Boulc'h, Laetitia, Nogry, Sandra, et Reffay, Christophe (dir.) (2024). « Introduction », dans *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 1-9. <https://doi.org/10.53480/2024iecare0bn/>



# **PARTIE 1**

## **ENSEIGNER L'INFORMATIQUE À L'ÉCOLE**



# Quel enseignement de l'informatique à l'école primaire en France ? Réflexions sur 40 ans de développements

# 1

Georges-Louis BARON<sup>1</sup>

1. Université Paris Cité, Laboratoire  
EDA, 75006 Paris, France

Dans l'enseignement français, l'informatique a pris différentes formes au fil du temps, souvent comme un ensemble d'outils pour l'enseignement et l'apprentissage des matières scolaires traditionnelles. Nous avons analysé en 2016 avec Béatrice Drot-Delange le cheminement suivi, le qualifiant de « tortueux et cahoteux » (Baron et Drot-Delange, 2016, p. 59). Les lignes qui suivent, rédigées en 2023, actualisent l'analyse. Elles exposent d'abord une synthèse sur l'apprentissage de la programmation en primaire depuis le début des années 1980 en reprenant des éléments factuels de cette publication. Puis elles proposent une réflexion sur l'évolution et les perspectives de la recherche en didactique de l'informatique, ainsi que sur les apports de la recherche participative en termes de problématisation de la question de l'informatique à l'école primaire.

## Un lent cheminement de l'informatique comme objet d'enseignement

Historiquement, l'idée que l'apprentissage de la programmation pouvait avoir des vertus éducatives est apparue dès la fin des années 1960, en particulier dans l'enseignement des mathématiques, avec pour outil privilégié le langage Logo (Feurzeig et Papert, 1968). Des expériences d'introduction de ce langage gérant un robot programmable (la tortue Logo) ont été menées en France dans les années 1970 et 1980 à l'école et au collège, avec un certain succès (Le Touzé *et al.*, 1979). Mais, jusqu'en 1979, l'ordinateur n'était guère entré que dans quelques dizaines d'établissements de second degré.



## **Au début de la décennie 1980, des politiques publiques pour l'enseignement primaire prêtant attention à la programmation**

Ce n'est qu'alors, avec le développement de la micro-informatique, que les autorités pédagogiques françaises ont lancé des politiques visant l'enseignement et favorisant l'industrie française. En effet à cette époque le soutien à l'industrie nationale de l'informatique était une considération importante pour les politiques. Les procédures d'acquisition étaient celles de marchés publics, suivant des cahiers des charges définis nationalement. Au début ce sont les lycées (où une expérimentation nationale avait été lancée en 1970) qui ont été les premiers bénéficiaires des politiques publiques, mais les autres ordres d'enseignement ont aussi été considérés.

Émilien Pélisset rappelle en 1985 qu'un plan d'équipement des écoles normales et de formation de leurs personnels avait été élaboré dès 1980. Il prévoyait l'équipement d'une école normale par académie sur quatre ans, accompagné d'une formation de six semaines pour une cinquantaine de leurs enseignants en 1981-82 et 1982-83.

Puis, un plan d'équipements en micro-ordinateurs « grand public » a été annoncé en 1983 à destination des écoles primaires. Il comportait notamment l'équipement des écoles de 16 départements avec des ordinateurs de fabrication française, des Thomson TO-7 relativement peu chers, utilisant un téléviseur standard comme écran, des cartouches et des cassettes magnétiques comme mémoire externe et gérant un crayon optique (ce qui était une nouveauté).

Cet auteur relève aussi la diversité des conditions d'implémentation des matériels et les insuffisances de la formation lors de cette opération.

Les formations s'effectuent de plus en plus hors du temps de service, pendant les congés. Mais surtout le TO-7 et les autres machines de l'Éducation nationale sont incompatibles ; le TO-7 ne fonctionne qu'avec un basic et des logiciels d'un éditeur vendu à la pièce et « protégés ». C'est la fin de la gratuité jusque-là assurée aux activités informatiques dans le service public. L'utilisation des logiciels de la bibliothèque du Centre national de documentation pédagogique (CNDP) est impossible. (Pélisset, 1985)

Deux mois après l'annonce de ce plan a paru une circulaire du directeur des écoles définissant des orientations pour ce niveau d'enseignement : l'informatique y est essentiellement considérée comme « un fait social », pour lequel il convient de mettre en place

une « sorte d'éducation civique ». En conséquence, elle doit être considérée dans la perspective d'activités d'éveil : « éveil humain et social, éveil technologique, éveil logistique »<sup>1</sup>. Concernant ce dernier aspect, la tortue Logo est mentionnée, ainsi que « la nécessité d'éveiller l'enfant à la pensée algorithmique, de lui faire saisir ce qu'est un programme (modestement bien sûr, au début, mais vraiment sur un véritable ordinateur) ».

Rappelons que le livre de Seymour Papert *Mindstorms : Children, Computers, and Powerful Ideas* venait d'être publié en français sous le titre *Jaillissements de l'esprit* (1981) et que le langage Logo suscitait un très fort intérêt, tant dans le public qu'à la Direction des écoles.

Deux ans plus tard, le Plan informatique pour tous (1985) a constitué une accélération et une inflexion considérables. Marqué par une vision de la responsabilité du système scolaire face au fait social informatique et par une intention homogénéisatrice, il a notamment réglé la question de l'équipement en ordinateurs des écoles, collèges et lycées publics en fournissant à tous une première dotation, dont le renouvellement incombait ensuite aux collectivités territoriales. La plupart des écoles primaires et des collèges ont reçu ce qu'on appelait des « nanoréseaux », c'est-à-dire des configurations de huit nanomachines Thomson autour d'un serveur PC. L'équipement ne se limitait pas aux ordinateurs et comportait aussi une dotation en logiciels à usage pédagogique (Baron et Bruillard, 1996).

Ce plan très ambitieux avait au moins un point faible. Les enseignants s'étaient vu offrir une formation d'une semaine pendant les vacances de Pâques, qui était cependant insuffisante pour espérer qu'ils puissent ensuite utiliser avec confiance les équipements disponibles face à des élèves. Un complément de formation avait probablement été prévu pour les années suivantes mais, suite à l'alternance politique de 1986, des orientations différentes ont été mises en œuvre<sup>2</sup> (Baron et Bruillard, 1996).

## **Depuis la fin des années 1980, occultation puis retour en grâce de l'informatique comme objet d'enseignement**

Depuis cette époque, l'intérêt institutionnel pour les opportunités ouvertes par l'apprentissage de la programmation (ou, comme on en est venu à dire plus récemment, du « codage ») a considérablement fluctué, on a connu une sorte de mouvement de balancier entre informatique outil et informatique objet d'apprentissage (Baron *et al.*, 2014).

Pour faire simple, on peut considérer que la décennie 1990-2000 a été celle d'un déni de l'intérêt de l'informatique comme objet

1. Dossier EPI n° 6 « l'informatique à l'école » (1984), <http://epi.asso.fr/revue/dossiers/d06p005.htm>

2. La gauche, au pouvoir depuis 1981, a perdu en 1986 la majorité à l'Assemblée nationale. Jacques Chirac est devenu premier ministre dans le cadre d'une cohabitation avec le président François Mitterrand.

de formation générale et d'un repli sur ce qu'on appelait « l'outil informatique ».

La décennie suivante a vu l'institutionnalisation de dispositifs d'évaluation des compétences en informatique comme le brevet d'informatique et Internet (B2i) pour les élèves) et le certificat informatique et Internet (C2i), pour les futurs enseignants. Ces certifications ont marqué une inflexion nette par rapport à la période précédente dans la mesure où, intégrées dans le système éducatif et gérées par le niveau national, elles indiquent la reconnaissance de la nécessité de compétences spécifiques en informatique, sans néanmoins prendre en considération ce qui relève de l'algorithmique et de la programmation et sans considérer la question des connaissances en jeu. Ces dernières sont timidement évoquées dans l'arrêté du 14 juin 2006<sup>3</sup>, qui indique de manière vague que les compétences à acquérir « résultent d'une combinaison de connaissances, de capacités et d'attitudes à mobiliser dans des situations concrètes ».

3.

<https://www.education.gouv.fr/bo/2006/29/MENE0601490A.htm>

La décennie 2010 a été celle de la montée d'un intérêt institutionnel pour l'informatique comme discipline, surtout après la publication d'un rapport de l'Académie des sciences (2013) au titre explicite : *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre* (Institut de France, Académie des sciences, 2013).

Pour le premier degré, ce rapport insiste sur la découverte de l'informatique par les élèves, soit en utilisant des ordinateurs, soit par l'intermédiaire d'activités débranchées. Il insiste aussi sur le fait que l'initiation à l'informatique est « une occasion pour les enfants de commencer à découvrir à un niveau élémentaire ce pan de notre culture, notamment écrite, fait de signes et de règles pour les manipuler » (p. 22).

Au cours de cette décennie 2010, les dispositifs de certification des compétences existants (B2i et C2i) ont connu un déclin. Ils ont été supplantés par Pix, dispositif original créé en 2016 comme une « startup d'État ». Cette expression, qui sonne comme un oxymore (comme l'indique le site gouvernemental qui présente ce dispositif, il ne s'agit pas d'une startup)<sup>4</sup>, désigne une initiative gouvernementale de 2013 destinée à « construire des services publics numériques ».

4. <https://labo.societenumerique.gouv.fr/fr/articles/les-startups-d'état-une-nouvelle-manière-de-construire-des-services-publics/>

5. Cette expression vise sans doute à indiquer une volonté de changement d'échelle et l'objectif d'accroître le chiffre d'affaires.

Une telle initiative est par essence à durée limitée. Pix est ensuite devenu, selon le site <http://www.pix.fr/>, une « scale-up »<sup>5</sup> offrant « un service public en ligne pour évaluer, développer, et certifier ses compétences numériques ».

Pix, qui n'est pas seulement destiné au système éducatif mais aussi au monde professionnel, est dédié à la certification. Son utilité possible comme outil de formation est cependant présente, comme l'indique le slogan « aider des millions d'utilisateurs à cultiver leurs compétences numériques tout au long de leur vie ».

Lavigne (2023), qui relève la place très réduite laissée à l'initiative enseignante, souligne l'accent mis sur la « gamification ». Cette dernière, classiquement, organise des parcours par niveaux avec indicateur de progression, jauges, récompenses virtuelles. L'auteur relève qu'en revanche il n'y a pas de marqueur social comme la position dans un classement. D'après lui, cela incite les utilisateurs à se focaliser sur l'augmentation de leur score personnel « au détriment du questionnement sur la réalité des phénomènes représentés » (p. 13).

S'agissant du second degré, le début de la décennie 2020 a (enfin) vu la création d'un certificat d'aptitude au professorat de l'enseignement du second degré (CAPES) puis d'une agrégation d'informatique, ce qui marque ainsi la « vraie » naissance de cette dernière comme discipline de formation générale de second degré, cinquante ans après la première politique nationale d'introduction de l'informatique au lycée et 40 ans après la demande explicite faite en 1980 par le rapport Simon au président de la République.

## **Quelle place pour la « pensée informatique » dans l'enseignement primaire : une question insistante**

Concernant le premier degré, ce qui s'est surtout diffusé dans les médias de masse comme dans le discours institutionnel relativement à l'intérêt de l'apprentissage de la programmation (ou du codage), c'est qu'il peut favoriser l'apprentissage d'un type de pensée spécifique, dont le nom a varié au cours du temps. Cela a d'abord été la démarche informatique (« algorithmique, organisatrice et modélisante ») qui a été promue par le colloque fondateur de Sèvres en 1970 (Baudé, 2018). Dans les années 1985, on a parlé de pensée logistique ou algorithmique. Puis l'idée de pensée informatique, introduite en 2006 par Wing (2006) s'est diffusée. Pour cette dernière, c'est une « forme de pensée analytique » (Wing, 2008) pour la résolution de problèmes, dont elle précise qu'elle ne nécessite pas forcément une machine et dont l'essence est « l'abstraction », ou plus précisément la gestion de différents niveaux d'abstraction.

Cette idée prend peu en considération d'autres aspects importants de l'informatique comme celui de la culture technique ou celui du champ « informatique et société ». Elle est donc réductrice par rapport au fait informatique en général.

En tout cas, l'idée de former à la pensée informatique (souvent considérée comme quasi équivalente au codage) a connu une certaine faveur dans la décennie 2010, en s'appuyant sur la large diffusion de nouveaux environnements, en particulier dans le domaine de la robotique pédagogique, avec les différents robots

de type Thymio ou BeeBot, ou les systèmes de programmation de type ScratchJr héritiers de Logo et adaptés aux jeunes enfants.

Ces environnements apparaissent désormais dans les manuels du primaire, en particulier de mathématiques (où ils viennent en appui à l'acquisition de notions de ce domaine). Ils interviennent aussi dans des actions d'innovation pédagogique. Ces dernières restent promues par des services ministériels et sont souvent soutenues par des collectivités territoriales (dont la responsabilité majeure porte sur la logistique de l'enseignement), ou par des opérateurs de type Canopé, ancien CNDP qui a récemment reçu une mission de formation continue des enseignants.

Au total, si l'informatique est devenue en 2020 une jeune discipline de second degré de plein exercice, elle n'a encore qu'une place relativement modeste dans les instructions officielles en tant que matière d'enseignement. Elle intervient principalement à l'école primaire sous la forme du codage, en proximité avec les mathématiques et pas de manière généralisée. L'évolution de la situation suppose que quelques problèmes difficiles reçoivent une solution : le premier est celui de la formation des enseignants au numérique en général et à la programmation en particulier ; le second, qui explique le précédent, est celui de la construction d'un consensus social sur les finalités de l'enseignement de l'informatique dans l'enseignement primaire.

## **Formation des maîtres et finalités de l'enseignement : deux problèmes importants et liés**

### **Former des maîtres : une nécessité critique**

Les politiques publiques des années 1970 et du début des années 1980 avaient accordé une grande importance à la formation en informatique des enseignants du second degré, investissant lourdement pour faire fonctionner un ensemble de centres de formation où des enseignants ont pu suivre une formation d'une année en étant totalement ou partiellement déchargés de cours.

Dans l'enseignement primaire, les anciennes écoles normales faisaient une place aux technologies nouvelles (surtout l'audiovisuel). Mais, depuis la création des Instituts universitaires de formation des maîtres (IUFM) en 1990, l'informatique et le numérique ont peiné à trouver une place, d'autant qu'à cette époque l'accent ne portait pas du tout sur la programmation, qui était même considérée comme quelque chose de plutôt indésirable face aux exigences du lire-écrire-compter.

En effet, c'est le numérique qui est venu au premier plan, avec ses nombreuses dimensions (Baron, 2018). D'autres sujets ont attiré l'attention des pouvoirs publics, comme la familiarisation avec une série d'instruments informatisés (au premier rang desquels le traitement de textes et de tableaux, où la programmation n'apparaît pas nécessaire<sup>6</sup>). On peut aussi citer l'éducation aux médias et à l'information (EMI), relevant en pratique de la responsabilité des enseignants documentalistes, ainsi que la culture technique, voire les humanités numériques.

Les formations d'enseignants, dépendantes des orientations ministérielles, prennent en conséquence peu en compte l'apprentissage de la programmation. Les formations continues, installées dans les différentes circonscriptions de l'enseignement primaire ont pris très partiellement le relais, avec une augmentation sensible du nombre de ressources en ligne disponibles. Certaines de ces ressources sont directement issues d'initiatives ministérielles comme *Prim à bord*<sup>7</sup> qui concerne le numérique à l'école en général et propose notamment des entrées comme codage / programmation, robotique et EMI. D'autres, ayant la forme de manuels, proviennent d'éditeurs (Vandevelde *et al.*, 2022). On trouve également des productions diverses venant d'associations, voire d'initiatives individuelles.

D'où provient la difficulté pour le système éducatif à offrir des formations en informatique aux enseignants du premier degré ? Dans un article fondateur, André Chervel (1988) observait que la création de disciplines scolaires dépend au premier chef d'un consensus social sur les finalités qui leur sont assignées. Cela a pris 50 ans (1970–2020) pour l'informatique dans le second degré.

## **Un déficit de consensus sur les finalités d'un enseignement d'informatique à l'école**

À l'évidence, la perspective d'une initiation à la pensée informatique ne fournit pas une base suffisante. D'autres propositions ont été faites, comme celles d'Eric Bruillard (2014). Pour celui-ci, il s'agit de transmettre une culture qui, « comme toute culture, a pour fondement une technicité partagée et valorisée ».

L'enjeu est de s'inscrire dans des visées larges, allant bien au-delà d'activités d'écriture de programmes (coder) dans l'objectif de commencer à maîtriser une technologie de travail : développer également des capacités d'action sur le monde et de compréhension de ce monde, des valeurs autour du travail individuel et collectif. (Bruillard, 2014)

6. Encore qu'on peut considérer que non seulement la création de macros mais aussi la création de formules dans un tableur relèvent déjà bien de la programmation. Il en va de même, dans une certaine mesure, de la définition des styles de paragraphes ou du publipostage dans un traitement de textes.

7. <https://eduscol.education.fr/196/prim-bord>

Sans doute, cela peut faire consensus, car personne ne peut ouvertement être contre l'idée de former des citoyens responsables. Mais comment avancer dans cette perspective ? Par quel moyen y parvenir ? Ne peut-on pour cela utiliser les matières et disciplines existantes ? C'est régulièrement le choix qui a été fait jusqu'ici au niveau primaire et la situation ne semble pas sur le point de devoir changer bientôt. Le ministère chargé de l'enseignement primaire ne semble en effet pas considérer l'initiation à l'informatique et au numérique comme une composante fondamentale de la culture générale, comme littérature spécifique.

Cette situation entraîne deux conséquences principales, non exclusives. La première est que les mathématiques, matière fondamentale, sont en passe d'acquiescer une sorte de prééminence voire d'exclusivité sur l'enseignement de l'informatique à l'école, avec comme corollaire une focalisation sur le codage d'algorithmes mathématiques. La seconde est l'intervention d'autres acteurs dans des cadres périscolaire et parascolaire : les familles, bien sûr, des associations de militants, mais aussi des collectivités territoriales et des opérateurs privés, éditeurs ou nouveaux fournisseurs de services éducatifs. Quels en seront les effets en termes de disparités sociogéographiques voire d'iniquités ?

## **Observer, documenter, problématiser, expliquer, une responsabilité importante pour la recherche**

### **Constitution progressive d'une communauté de didactique de l'informatique s'intéressant à l'enseignement primaire**

Des recherches francophones sur l'apprentissage de la programmation par des jeunes d'école primaire ont été menées dès la fin des années 1970 dans la dynamique de la mouvance Logo (Robert, 1985), produisant des résultats assez convergents : dans des conditions favorables, les jeunes sont capables de réaliser des programmes assez complexes. Mais les limites de l'initiation à une pensée procédurale ont également été soulignées, tout comme la difficulté à observer des transferts d'apprentissage dans d'autres domaines (Crahay, 1987).

Au niveau international, des recherches sur l'apprentissage de l'informatique utilisant les nouveaux environnements robotiques ou de type Scratch Jr n'ont pas cessé de se développer, en particulier aux États-Unis (Bers, 2010 ; Wilson et Moffat, 2010). Il est à noter qu'en Europe, des chercheurs grecs ont aussi été précurseurs (Komis et Misirli, 2012).

En France même, une communauté s'est constituée en s'appuyant sur un noyau de chercheurs s'intéressant à l'informatique dans l'enseignement primaire. Des thèses ont été soutenues dès la fin des années 1990 (comme Harrari, 2000); des supports de publication ont été ouverts aux jeunes chercheurs (en particulier la revue *adjectif.net*) et différents types de colloques servant à diffuser les idées et à en susciter de nouvelles ont été créés.

Parmi ces derniers, il convient de mentionner la série *Didactique des progiciels* (DIDAPRO) lancée à l'université de Créteil en 2003 lors d'une période d'étiage de l'intérêt institutionnel pour l'enseignement de l'informatique, renommée en 2011 lors de sa quatrième édition à Patras (Grèce) DIDAPRO-DIDASTIC pour marquer son intérêt pour les sciences et technologies de l'information et de la communication<sup>8</sup>. C'est d'ailleurs à partir de cette édition qu'on commence à voir un nombre significatif de contributions relatives à l'école primaire.

8. <https://www.didapro.org>

Par ailleurs, la série de conférences *École et TIC* (ETIC), lancée en 2013 à l'université de Limoges à l'initiative de Béziat et Villemonteix<sup>9</sup> et spécialement focalisée sur l'école primaire, a également joué un rôle important de regroupement d'une communauté. Un élément central de cette dynamique a été l'engagement de chercheurs issus de toute l'aire francophone et même au-delà (Nogry *et al.*, 2019).

9. <https://colloque-etic-4.sciencesconf.org/>

Vers le milieu de la décennie 2010, l'intérêt croissant des pouvoirs publics pour l'informatique comme objet possible d'enseignement à l'école primaire n'a sans doute pas été étranger à la suite favorable donnée à certains projets de recherche soumis à l'Agence nationale de la recherche (ANR).

Le projet Didactique et apprentissage de l'informatique à l'école (DALIE) (2016–2018)<sup>10</sup> a d'abord visé à faire le point sur les représentations que les enseignants, les élèves ont de l'informatique, sur les représentations que les étudiants se préparant à l'enseignement primaire et leurs formateurs se font de la place à donner aux technologies informatisées à l'école. Il a produit des résultats de recherche sur l'apprentissage de l'informatique à l'école primaire, fédéré des initiatives de recherche sur le territoire et préparé le terrain pour un projet se fondant sur ces premiers acquis : « Informatique à l'école : conceptualisations, accompagnement, ressources » (IE CARE), qui lui a succédé en 2018.

10. <https://www.unilim.fr/dalie/le-projet/>

## IE CARE, projet producteur de connaissances

L'objectif principal de cette dernière recherche a été de réfléchir de manière pluridisciplinaire à des modalités durables pour un enseignement de l'informatique à l'école obligatoire. Trois directions principales ont été explorées : délimiter un ensemble de



contenus informatiques enseignables ; concevoir et tester l'usage de scénarios pédagogiques et de ressources pour soutenir des pratiques d'enseignement et d'apprentissage de l'informatique, et mettre en place un cadre d'accompagnement pour les enseignants et les formateurs en informatique.

Parmi les nombreux enseignements qu'on peut tirer de cette recherche, je retiens principalement ici qu'elle confirme les premiers résultats obtenus lors des expérimentations Logo. Même de jeunes élèves sont capables de se familiariser avec des dispositifs informatisés et de réaliser des projets assez complexes, pourvu qu'ils soient guidés par une programmation didactique bien pensée (Touloupaki, 2023). En mettant en œuvre de la programmation, les jeunes n'apprennent sans doute pas tant des concepts informatiques qu'ils n'acquièrent les premiers rudiments d'une culture technique et se familiarisent avec des instruments et des conceptualisations du fonctionnement de systèmes matériels et logiciels. Ce qui restera dans leur esprit dépend de la récurrence de ce type d'expérience, qui peut se produire non seulement au sein de l'enseignement formel mais aussi lors d'activités menées dans des cadres périscolaires et extrascolaires.

Le projet a aussi confirmé qu'il est nécessaire pour les enseignants de bénéficier de conditions favorables en termes de ressources (pas seulement techniques d'ailleurs mais aussi institutionnelles). Une maîtrise de l'informatique en général ne leur est pas nécessaire. Il importe en revanche qu'ils soient sûrs de leur pédagogie, qu'ils disposent du point de vue de l'informatique d'une familiarité suffisante avec les dispositifs utilisés et qu'ils aient une idée claire des conceptualisations à susciter. Une difficulté est que les activités à conduire en classe ne sont pas encore bien alignées avec le fonctionnement traditionnel de l'école. Elles représentent une innovation pédagogique nécessitant de gérer de nouveaux aléas et sont assez coûteuses en investissement personnel, ce qui gêne leur banalisation rapide.

## **Quelles perspectives ?**

L'enseignement scolaire restant la base sur laquelle se construit une culture commune, une question insistante est alors celle d'une prise en compte plus nette de l'informatique dans les instructions officielles et dans les programmes d'étude de l'enseignement obligatoire, ce qui sort clairement du cadre de responsabilité des chercheurs.

Ces derniers ont néanmoins la responsabilité de documenter les initiatives prises par les praticiens (et donc de les observer, en accord avec eux), d'identifier des problèmes posés (en particulier

lors des sorties de phases expérimentales), de comprendre ce qui se joue, de diffuser leurs idées pour que d'autres s'en emparent.

L'intérêt pour la didactique de l'informatique (et plus largement du numérique) devrait se maintenir encore un moment et se diriger aussi dans les années à venir vers ce qu'il est possible de transmettre lors des premières années de collège dans le domaine de la culture technique et de la mise en œuvre de concepts proprement informatiques, en considérant aussi les modalités non formelles d'apprentissage fondées sur la recherche de ressources en ligne et sur l'entraide entre pairs.

Dans la mesure où il s'agit de documenter des situations, de les problématiser, de les comprendre, la recherche participative, associant différents types d'acteurs, se révèle une modalité utile.

Elle pose cependant des problèmes de différents ordres. Par exemple, le projet IE CARE avait bel et bien prévu de privilégier cette perspective, notamment pour étudier certaines pratiques d'enseignement ainsi que la mise en œuvre de formations. Malheureusement, la crise sanitaire de la Covid-19 a imposé l'annulation de séances de formation continue où il était prévu de développer des collaborations avec des praticiens. De plus, il a été pratiquement impossible de se rendre dans les classes pendant une période assez longue, ce qui a empêché d'enclencher une dynamique. Seuls les terrains où des liens de confiance entre chercheurs et praticiens avaient pu être établis avant le projet ont pu suivre ce mode de fonctionnement.

Même quand les conditions pratiques sont favorables, la situation peut néanmoins se compliquer lorsque sont explorées des directions qui ne sont pas assez en phase avec les orientations officielles du moment : les décideurs soutenant le projet peuvent alors exprimer une frustration qui peut avoir des conséquences notables quand ils ont une responsabilité de supervision sur les praticiens souhaitant participer à la recherche.

On est alors face à une situation très classique, illustrant à nouveau les rapports souvent difficiles entre savants et politiques. Sa gestion fait intervenir de la diplomatie et des rapports de force où la position des chercheurs seuls est plutôt faible. Ce qui nourrit pratiquement cette position, c'est moins leur capacité à apporter des résultats généraux qu'à proposer des réflexions étayées par des observations menées selon les règles de l'art et à les publier.

Comme nous le soulignons dans un texte écrit avec Cédric Fluckiger traitant du problème plus large de la recherche sur les technologies (Baron et Fluckiger, 2021), un enjeu est le développement dans la durée de forums d'échanges pluridisciplinaires permettant de constituer des collectifs hybrides durables intéressés par la recherche participative.

## Références

- Baron, G.-L. (2018). *Informatique et numérique comme objets d'enseignement scolaire en France : entre concepts, techniques, outils et culture*, Didapro 7 – DidaSTIC. De 0 à 1 ou l'heure de l'informatique à l'école. <https://hal.archives-ouvertes.fr/hal-01760479/document>
- Baron, G.-L., et Bruillard, É. (1996). *L'informatique et ses usagers dans l'éducation*, PUF. [http://mutatice.net/glbaron/lib/exe/fetch.php/baron\\_bruillard\\_livre1996-fprepub2.pdf](http://mutatice.net/glbaron/lib/exe/fetch.php/baron_bruillard_livre1996-fprepub2.pdf)
- Baron, G.-L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie. Recherches en éducation*, vol. 195, n° 2, p. 51-62. <https://doi.org/10.4000/rfp.5032>
- Baron, G.-L., et Fluckiger, C. (2021). Approches et paradigmes pour la recherche sur les usages éducatifs des technologies : enjeux et perspectives, *Canadian Journal of Learning and Technology*, vol. 47, n° 4. <https://doi.org/10.21432/cjlt28059>
- Baron, G.-L., Drot-Delange, B., Grandbastien, M., et Tort, F. (2014). Computer science education in French secondary schools : Historical and didactical perspectives, *Trans. Comput. Educ.*, vol. 14, n° 2, p. 1-27. <https://doi.org/10.1145/2602486>
- Baudé, J. (2018). *Le séminaire de Sèvres (mars 1970)*, *Revue de l'EPI*, n° 201.
- Bers, M. U. (2010). The tangible K robotics program : Applied computational thinking for young children, *Early Childhood Research & Practice*, vol. 12, n° 2. <http://eric.ed.gov/?id=EJ910910>
- Bruillard, É. (2014). *Une voie pour penser et construire une formation à l'informatique pour les élèves de l'école primaire ?* <https://hal.science/hal-03948939>
- Chervel, A. (1988). L'histoire des disciplines scolaires. Réflexions sur un domaine de recherche, *Histoire de l'éducation*, vol. 38, n° 1, p. 59-119. <https://doi.org/10.3406/hedu.1988.1593>
- Crahay, M. (1987). Logo, un environnement propice à la pensée procédurale, *Revue française de pédagogie*, vol. 80, n° 1, p. 37-56. <https://doi.org/10.3406/rfp.1987.1473>
- Feurzeig, W., et Papert, S. (1968). « Programming-languages as a conceptual framework for teaching mathematics » dans OTAN, *Programmé / tendances actuelles, Actes d'un colloque OTAN*, p. 233-246.
- Harrari, M. (2000). *Informatique et enseignement élémentaire 1975-1996. Contribution à l'étude des enjeux et des acteurs*, thèse de l'université René Descartes. <https://tel.archives-ouvertes.fr/edutice-00000406/document>

INRP (1981). *Dix ans d'informatique dans l'enseignement secondaire. 1970-1980*, Institut national de recherche pédagogique.

[https://www.epi.asso.fr/blocnote/Dix\\_ans\\_INRP\\_1981.pdf](https://www.epi.asso.fr/blocnote/Dix_ans_INRP_1981.pdf)

Institut de France, Académie des sciences (2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*, Académie des sciences. [http://www.academie-sciences.fr/pdf/rapport/rads\\_0513.pdf](http://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf)

Komis, V., et Misirli, A. (2012). Jeux programmables de type Logo à l'école maternelle, *Adjectif : analyse et recherche sur les TICE*. <http://www.adjectif.net/spip/spip.php?article140&lang=fr>

Lavigne, M. (2023). Formes et sens de l'innovation éducative gamifiée. Une étude de cas : La plateforme Pix, *Technologie et innovation*, vol. 8, n° 3. <https://doi.org/10.21494/ISTE.OP.2023.0972>

Le Touzé, J.-C., N'gosso, I., Robert, F., et Salamé, N. (1979). *Apports d'un environnement informatique dans le processus d'apprentissage. Projet LOGO*, Institut national de recherche pédagogique. Département de recherche sur les applications éducatives des technologies de communication, section « informatique et enseignement ».

Nogry, S., Boulc'h, L., et Villemonteix, F. (2019). *Le numérique à l'école primaire : pratiques de classe et supervision pédagogique dans les pays francophones*, Presses universitaires du Septentrion.

Pélisset, É. (1985). Pour une histoire de l'informatique dans l'enseignement français – Premiers jalons, *Système éducatif et révolution informatique*. [Republié dans la *Revue de l'EPI (Enseignement Public et Informatique)*], vol. 50.

<https://www.epi.asso.fr/revue/histo/h85ep.htm>

Robert, F. (1985). L'utilisation de l'ordinateur dans l'enseignement primaire : l'exemple de la France, *Enfance*, vol. 38, n° 1, p. 19-30.

<https://doi.org/10.3406/enfan.1985.2857>

Simon, J. C. (1980). *L'éducation et l'informatisation de la société : Rapport au président de la République*, Documentation française.

Touloupaki, S. (2023). *Contribution à l'étude de l'apprentissage de la programmation en grande section et en cours préparatoire, à travers le logiciel ScratchJr : Une approche didactique exploratoire*, thèse de l'université Paris Cité et de l'université de Patras. <https://theses.fr/s177034>

Vandeveldel, I., Fluckiger, C., et Nogry, S. (2022). Resources and textbooks for computer science education in French primary schools, *IARTEM E-Journal*, vol. 14, n° 1, p. 01-20. <https://doi.org/10.21344/iartem.v14i1.954>

Wilson, A., et Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming, *Proceedings of the 22nd*

*Annual Workshop of the Psychology of Programming Interest group-PPIG2010.* <https://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf>

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

Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, vol. 366, n° 1881, p. 3717-3725. <https://doi.org/10.1098/rsta.2008.011>

Pour citer ce chapitre :

Baron, Georges-Louis (2024). « Quel enseignement de l'informatique à l'école primaire en France? Réflexions sur 40 ans de développements », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 13-26. <https://doi.org/10.53480/2024iecare01o/>



# Quel enseignement de l'informatique dans la scolarité obligatoire en France ? Analyse des programmes et manuels

## 2

Isabelle VANDEVELDE<sup>1</sup>  
Cédric FLUCKIGER<sup>1</sup>  
Olivier GRUGIER<sup>2</sup>  
Mariam HASPEKIAN<sup>2</sup>

1. Université de Lille, ULR 4354, CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, 59000 Lille, France

2. Université Paris Cité, Laboratoire EDA, 75006 Paris, France

Les disciplines scolaires sont des ensembles instables « objet de constructions, au travers de luttes, de compromis et d'adaptations » (Reuter et Lahanier-Reuter, 2007). Elles font l'objet de « constantes reconfigurations » (Harlé, 2016). Ainsi, depuis 2015, un enseignement de l'informatique est prévu en primaire et au collège, sans y être néanmoins identifié comme une discipline à part entière, en même temps que de nouvelles options et spécialités ont été introduites au lycée. En outre, l'enseignement de l'informatique est inscrit dans le socle commun de connaissances, de compétences et de culture (S4C).

Comme c'est le cas pour toute introduction de contenus dans les programmes scolaires français, ceux liés à l'informatique passent par des textes prescriptifs avant de se retrouver transposés dans des manuels destinés aux enseignants et élèves. Tous ces documents, relevant des prescriptions et recommandations de l'enseignement de l'informatique et de leurs transpositions dans les manuels, forment un discours sur cet enseignement de l'informatique, qui sert de référence et oriente les pratiques des enseignants. Ce chapitre vise à décrire et analyser cet espace scriptural, à montrer comment les instructions officielles, les discours de l'institution scolaire ou encore les manuels scolaires parlent de l'informatique et de son enseignement. Ce tableau, brièvement esquissé dans ce chapitre, permet en effet d'interpréter, dans les chapitres suivants, ce que les acteurs font effectivement ou comment ils se font une idée de ces enseignements et apprentissages. Qu'est-ce que ces textes prévoient d'enseigner aux élèves ? Dans quels buts ? Selon quelles modalités ? Comment ces contenus sont-ils désignés et délimités ? L'enseignement de l'informatique prévu en 2015 rompt-il avec les enseignements passés de l'informatique, fragmentés au sein d'autres matières scolaires (Baron et Drot-Delange, 2016) ?

Pour répondre à ces questions, nous croisons des apports théoriques et méthodologiques issus de la didactique des disciplines francophones à ceux développés par des chercheurs en informatique, pour comprendre et caractériser les contenus informatiques qui sont enseignés à l'école.

Après avoir présenté le cadre théorique puis la méthodologie employée pour sélectionner et analyser ces textes, nous analyserons la manière dont l'informatique est perçue au sein de notre corpus de textes. Pour cela, nous nous appuyerons sur les catégories d'analyse des disciplines proposées par Reuter (2014).

## **Cadre théorique : analyser l'informatique scolaire comme une discipline**

Si, dans une acception large on peut entendre par informatique scolaire « l'ensemble des manifestations des technologies informatiques ou de la science informatique dans le champ scolaire » (Fluckiger, 2019a), nous ne nous intéressons ici qu'à ce que l'on peut identifier comme des contenus qui peuvent être rattachés au domaine de l'informatique au sein d'une discipline scolaire d'enseignement. Les trois termes, informatique, contenu et discipline méritent une clarification théorique.

À suivre Bruillard (2014), il est difficile de s'accorder sur une ontologie de l'informatique, qui est à la fois une science, une technologie mais aussi un ensemble d'usages sociaux. De fait, l'école peut enseigner des notions directement issues de la science informatique, comme la notion de boucle ou de variable, mais les élèves sont aussi confrontés à des contenus moins directement reliés à ce que des informaticiens peuvent considérer comme relevant de leur discipline de recherche : allumer un ordinateur, taper au clavier, identifier les dangers liés aux réseaux sociaux. . . Dans ce chapitre, nous cherchons à comprendre la logique et l'articulation de ces différents contenus dans un ensemble de textes, dans lesquels nous cherchons à repérer tout ce qu'on peut qualifier de contenu.

Le terme de contenu, en didactique, permet de désigner de manière très large ce qui fait l'objet d'un enseignement ou d'un apprentissage (Daunay *et al.*, 2015). Comme le propose Delcambre (2013), nous considérons en effet que les didactiques s'intéressent non seulement aux savoirs, mais aussi aux savoir-faire, savoir-être, valeurs, rapports à, etc. Ainsi, nous chercherons à repérer, dans un corpus de textes, ce que nous pouvons identifier comme un contenu informatique, dans ces différentes dimensions.

Ces textes définissent une organisation des contenus visant leur enseignement. C'est dans ce sens que nous parlerons de discipline

scolaire pour désigner, à la suite de Reuter, « de manière minimale, un mode d'organisation, à des fins d'enseignements et d'apprentissages, de contenus, de dispositifs, de pratiques, d'outils... » (Reuter, 2014). Nous considérons que ce mode d'organisation des contenus informatiques peut être analysé, à l'instar des disciplines scolaires constituées. C'est pourquoi, mobilisant les catégories d'analyse proposées par Reuter (2014), nous examinerons successivement le fonctionnement institutionnel et la désignation des contenus informatiques, la variété et les types de contenus, le fonctionnement pédagogique-didactique, les modes de références aux discours théoriques et les visées assignées à ces contenus à l'école primaire. Ces éléments seront plus précisément expliqués dans les sections correspondantes.

Pour comprendre finement les pratiques et représentations des apprenants comme des enseignants, qui font l'objet des chapitres suivants de cet ouvrage, il est nécessaire de les situer dans le cadre des prescriptions et recommandations qui définissent ce que les enseignants sont censés faire dans ce domaine avec leurs élèves. Ces prescriptions et recommandations constituent un espace scriptural, définissant l'enseignement de l'informatique et formant, avec l'espace des pratiques de la discipline et celui des représentations, ce que Reuter nomme la configuration disciplinaire (Reuter, 2004 ; 2007). L'un d'entre nous a déjà proposé de considérer ces prescriptions, pour comprendre le contexte d'enseignement et d'apprentissage de l'informatique (Fluckiger, 2011 ; Fluckiger et Reuter, 2014).

## Méthodologie : constitution du corpus

Afin d'établir les catégories de contenus en jeu dans l'enseignement de l'informatique, nous analysons un corpus constitué de trois ensembles de textes :

- les programmes scolaires de 2015, en vigueur au moment de notre analyse, pour les cycles 1-2-3-4 de l'école française (élèves âgés de 2 à 14 ans) : ministère de l'Éducation nationale (MEN, 2015a ; 2015b) pour les différents cycles ;
- un ensemble de textes institutionnels évoquant le numérique ou l'informatique dans l'éducation, qui constitue un contexte de la mise en place des enseignements de l'informatique ainsi que des indications des intentions des auteurs des prescriptions ;
- des manuels d'informatique pour l'école élémentaire (identifiés ici par un numéro).



Les programmes sont tous analysés. En revanche, les autres textes institutionnels et les manuels, trop nombreux pour être tous analysés, ont fait l'objet d'une sélection.

## Sélection des textes institutionnels

Notre corpus regroupe une sélection de textes institutionnels : des rapports émis par les parlementaires ou par des organismes proches des décideurs ministériels. Ces textes nous semblent représentatifs des conceptions de ceux qui sont à l'origine de l'introduction de l'informatique dans les programmes. Un tel corpus ne pouvant être exhaustif faute de définition précise de ce qui constitue les discours de la « noosphère »<sup>1</sup>, nous avons fait le choix de limiter notre analyse à cinq documents datés de 2014 à 2018, car ils ont été largement diffusés et cités (liste en annexe).

1. La noosphère désigne, selon Chevallard (1985), « la sphère où l'on pense – selon des modalités parfois fort différentes – le fonctionnement didactique ».

## Sélection des manuels d'informatique

Un ensemble de dix ouvrages d'enseignement de l'informatique est analysé. Ces ouvrages sont destinés aux élèves des cycles 1 à 4. Les dix ouvrages se distribuent ainsi : un ouvrage pour le cycle 1 et trois ouvrages pour chacun des autres cycles.

Une multitude d'ouvrages d'informatique destinés aux élèves sont disponibles sur le marché (au moment de la constitution du corpus, 217 manuels scolaires d'informatique étaient disponibles sur le site d'un libraire populaire). Pour ne conserver qu'un nombre limité de manuels sans faire intervenir notre propre subjectivité, nous avons mobilisé deux critères. Premièrement, les ouvrages devaient être édités à partir de 2015, année de mise en place des programmes scolaires analysés. Deuxièmement, le titre, la couverture ou quatrième de couverture des ouvrages analysés devaient renvoyer explicitement à l'âge, au niveau de classe ou au cycle des élèves, car ce critère est un critère majeur du choix de manuel par les enseignants qui veulent les utiliser en cours. Ces critères ont permis d'identifier dix manuels répartis sur tous les cycles.

Dans ces ouvrages, nous avons repéré des unités élémentaires (exercice ou page de cours), que nous avons ensuite classées en fonction du type de contenu abordé. Nous disposons donc d'un corpus de 1170 unités : 730 pages de cours et 440 exercices.

La suite de l'article présente les résultats de nos analyses de ce corpus, programmes, textes institutionnels et manuels, selon les catégories d'analyse de la discipline proposés par Reuter (2014), exposées plus haut.

## Fonctionnement institutionnel : désignation des contenus informatiques

Par « fonctionnement institutionnel », Reuter (*ibid.*) entend à la fois les désignations de la discipline, ses modes de présence (en particulier sa permanence), son « poids » en termes d'horaires, ses coefficients, etc.

Après une première introduction dans les années 80, puis disparition, depuis maintenant sept années, un enseignement de l'informatique a été réintroduit dans les programmes de l'école primaire française, montrant une volonté politique de construire une culture commune pour tous dans ce domaine. Les textes, cependant, ne constituent pas un support permettant de clarifier, pour les enseignants, ce qui doit être enseigné.

Ainsi, la question de la désignation de l'informatique soulève des difficultés et permet de déplier certains implicites des promoteurs de ces contenus dans les programmes. En effet, les termes informatique et numérique désignent des ensembles mal délimités, qui se recouvrent parfois très largement. Pourtant, ces deux termes renvoient à des réalités distinctes et complémentaires (Baron et Boulc'h, 2012). Les mots polysémiques pour décrire les attendus de l'institution peuvent alors apporter des confusions pour des enseignants pas toujours suffisamment formés.

### Dans les textes institutionnels

Ce n'est pas le rôle des textes institutionnels que de définir précisément les contenus. Ils rendent compte de la vision plus générale qui préside à la rédaction des programmes. Dans ces textes, le numérique, utilisé comme nom, est supposé être « porteur de nombreuses opportunités » (texte 5) et « pallie[r] les défaillances de notre système éducatif » (texte 3). Ce discours s'inscrit dans les mythes du numérique en éducation largement démentis sur un supposé pouvoir transformatif du numérique (Amadiou et Tricot, 2014; Fluckiger, 2019b; Livingstone, 2012). Le terme numérique, utilisé comme adjectif, fait référence au « domaine du numérique » (texte 4), comme dans l'expression « vie numérique » (texte 5), pour lequel il est nécessaire que les élèves développent des compétences car « le quotidien des enfants est déjà numérique » (texte 3). Dans plusieurs textes, « numérique » est utilisé à la fois comme substantif et comme adjectif, parfois dans la même phrase, par exemple : « les enseignements portant à la fois spécifiquement sur le numérique ou utilisant des ressources et outils numériques » (texte 4).

Le terme numérique peut renvoyer à une supposée nouvelle forme de rapport au savoir remettant en cause la place même

de l'éducation : « Le savoir change, ses modes de transmission et notre rapport à celui-ci également. Ce dernier échappe désormais au monopole des institutions académiques traditionnelles » (texte 3).

Le terme numérique est largement associé à une idée de changements brutaux et importants : « révolution numérique », « transformation numérique », « bouleversements numériques » sont des syntagmes qui reviennent avec constance sous la plume des acteurs institutionnels. Ainsi, l'idée qui se dégage est celle d'une évolution qu'il est nécessaire d'accompagner, le numérique représentant à la fois un défi et une opportunité.

Dans toutes ces acceptions, il est moins question d'une forme de pensée informatique (Wing, 2006) que de l'utilisation d'instruments (notamment pour enseigner) et d'apprendre aux élèves à les utiliser, idée souvent exprimée dans le vocable, à la mode, des compétences.

## Dans les prescriptions officielles

Ce qui est marquant à la lecture des programmes d'enseignement et du S4C, est l'enchevêtrement des termes numérique et informatique. Si le S4C semble les distinguer au travers de deux domaines différents, les programmes d'enseignement utilisent ces deux termes sans réelle distinction.

L'informatique se retrouve dans le premier domaine d'enseignement du S4C, relatif à l'acquisition du langage. Dans ce domaine, les langages sont à la fois perçus comme un objet d'apprentissage et comme un outil d'apprentissage. L'enseignement de l'informatique correspondrait alors à l'acquisition et à l'utilisation d'un langage spécifique. Le S4C précise trois objectifs relatifs à l'utilisation des langages informatiques :

[L'élève] sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. (S4C)

Au sein des programmes des cycles 1 à 4, nous recensons 93 éléments relatifs à l'enseignement de l'informatique : 53 relèvent des « attendus de fin de cycle [et les] connaissances et compétences associées » (les attendus) et 40 des « exemples de situations, d'activités et de ressources pour l'élève » (les exemples d'activité).

Pour le cycle 2, le mot numérique est employé aussi bien comme adjectif qualificatif pour préciser une technologie (« outils numériques ») que pour s'opposer à un geste physique (« manuscrite ou numérique ») pour évoquer l'apprentissage du traitement de texte). Alors que le traitement de texte est nommé « dispositif informatique » dans ce même texte prescriptif.

## Dans les manuels

Presque tous les ouvrages du corpus renvoient sur leur couverture à la programmation au travers de termes ou d'expressions tels que « à la découverte du codage », « je code », « apprends à programmer », etc.

Les ouvrages du corpus ont une forte tendance à associer l'informatique à l'apprentissage d'un langage, du codage et de la programmation, se plaçant ainsi dans la continuité des textes prescriptifs et du S4C.

Au final, dans nos corpus textuels, les contenus relatifs à l'informatique constituent un groupe qui est loin d'être homogène. Les contenus informatiques repérés ici sont de natures diverses. Par exemple, les attendus issus des programmes d'enseignement « copier à l'aide d'un clavier » (MEN, 2015a), « décrire l'architecture simple d'un dispositif informatique » (MEN, 2015b) ou encore « écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné » (MEN, 2015b), ne sont pas du même ordre et illustrent le fait que l'informatique est un domaine large recouvrant plusieurs champs.

## Fonctionnement institutionnel : autonomie de la discipline

Au-delà de la désignation des contenus, le fonctionnement institutionnel renvoie à l'autonomie de la discipline. Les textes institutionnels oscillent entre deux positions : celle d'une discipline autonome, qui est réservée au lycée à travers la création d'options et de spécialités et la position, dominante, d'un enseignement au sein des disciplines existantes, comme cela était le cas pour le brevet informatique et Internet (B2i, cf. Fluckiger et Bart, 2012). Ainsi, le Conseil national du numérique (CNUM) indique que l'informatique « se met en œuvre dans toutes les disciplines » (p. 20) ou de la « nécessité d'initier les lycéens à une démarche de pensée "nouvelle, modélisante, algorithmique et organisationnelle", traversant toutes les disciplines » (p. 22) et précise que « l'enseignement par discipline de l'Éducation nationale se prête mal à cette nécessité d'approche globale » (p. 25).

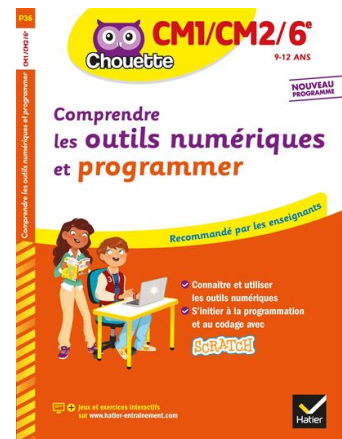


FIGURE 1 – Première de couverture d'un manuel du corpus (manuel 5 : Cohen, A., et Marcialis, J. (2018). *Comprendre les outils numériques et programmer*). © Éditions Hatier, 2020.

Tout d'abord, l'école primaire en France se caractérise par la non spécialité disciplinaire des enseignants et par des programmes articulant des contenus de plusieurs disciplines. En outre, les organisations curriculaires de l'école primaire ne sont pas tout à fait des disciplines scolaires au sens qu'elles prennent dans le secondaire. Ainsi, le temps alloué aux enseignements de l'informatique n'est pas précisé dans les programmes scolaires. On peut cependant dire que selon ces critères, l'informatique scolaire est de toute évidence une matière instable, peu présente en termes horaires, et pas toujours mise en œuvre (pour un historique, voir le chapitre 1 « Quel enseignement de l'informatique à l'école primaire en France ? Réflexions sur 40 ans de développements »).

Dès le collège, les attendus concernant les notions informatiques se retrouvent disséminés, au sein de deux champs disciplinaires, les mathématiques et la technologie, rendant la recherche d'informations complexe et incertaine, d'autant qu'il est difficile de savoir si un texte donné est toujours en vigueur ou s'il a été mis à jour ultérieurement. Les instructions officielles se découpent en :

- un programme par cycle de trois années (cycle 3 : cours moyen 1<sup>re</sup> et 2<sup>e</sup> année (CM1, CM2), 6<sup>e</sup>, et cycle 4 : 5<sup>e</sup>, 4<sup>e</sup>, 3<sup>e</sup>), découpé en plusieurs compétences, mis à jour en 2018 puis en 2020 ;
- un socle dit « commun de connaissances, compétence et culture », découpé en quatre domaines ayant chacun plusieurs composantes (BO n°17 du 23 avril 2015) ;
- et divers documents d'accompagnement : des documents ressources dédiés à chacun des thèmes du programme (dont un document d'accompagnement pour le thème « Algorithmique et programmation »), des documents « Repères de progression et attendus de fin d'année », publiés au Bulletin officiel (BO) n°22 du 29 mai 2019, enfin un « document d'accompagnement pour l'évaluation des acquis du socle commun ».

L'analyse de ces prescriptions montre une présence de l'informatique ainsi éparpillée en compétences, thèmes, connaissances, domaines et composantes, ce qui n'aide pas à clarifier les attentes institutionnelles liées à cet enseignement. Par ailleurs, l'analyse des prescriptions en mathématiques montre que l'informatique s'y présente de manière floue (Amadiou et Tricot, 2014), avec des éléments qui ne s'affirment pas comme légitimement ancrés en mathématiques. Pour exemple, dans le programme du cycle 3, la référence à l'informatique, à la programmation et à l'algorithmique est située dans le « domaine 2 : Les méthodes et outils pour apprendre » alors que l'initiation à la programmation y est préconisée, au milieu des savoirs en mathématiques, à partir

d'activités géométriques et que l'annexe « repères annuels de progression pour le cycle 3 » (*ibid.*) montre une visée plus large. En lien avec ces savoirs sur la programmation et l'algorithmique, dans la partie des programmes de sciences et technologie du cycle 3, il est aussi prescrit que les élèves doivent être capables d'« appliquer les principes de l'algorithme et de la programmation par blocs pour écrire ou comprendre un code simple ». Au cycle 4, le terme « algorithme » est présent en lien avec la compétence « communiquer » (MEN, 2020, Annexe 3, p. 130) qui est peut-être la compétence la moins spécifique aux mathématiques, les autres étant « chercher, modéliser, représenter, raisonner, et calculer » dans lesquelles « algorithme » aurait tout à fait eu sa place. Ici, on trouve en outre des expressions telles que : « dans une démarche de projet », « développent des méthodes de programmation » . . . qui ont peu de sens pour des enseignants en mathématiques non formés ni à la démarche de projet, ni à des méthodes de programmation.

Enfin, un dernier trait saillant de ces programmes est une évolution qui va paradoxalement vers une diminution de la place occupée par l'informatique au fil du temps. Alors que la place de l'informatique et des technologies numériques ne cesse de s'accroître dans nos sociétés, les programmes de 2020 montrent une baisse de la place

**TABLEAU 1** – Présence des termes liés au champ de la programmation informatique dans les programmes de mathématiques de cycle 4; de 2008 à 2020.

	2008	2015	2018 et 2020
Informatique / salle info	10	6	4
Algorithmique	0	3	3
Algorithme(s)	4 mais hors contexte de programmation	2 (dont un dans un contexte explicite de programmation)	2 (dont un dans un contexte explicite de programmation)
Coder / code	0	6 dont un en lien avec la programmation (code ASCII)	4 dont 0 en lien avec la programmation
Codage	0	0	0
Programmer / programme (au sens informatique)	0	11	8
Programmation	0	6 (dont programmation événementielle)	5 (programmation événementielle disparaît)
Logiciel (dont géométrie dynamique)	21 (15)	14 (8)	7 (4)
Logiciel (autre que de géométrie dynamique)		6	3
Instruction (en lien avec l'informatique)		3	2
Stockage de l'information		3	1
Robot	0	1	0
Ordinateur	9	1	1

accordée à l'algorithmique, la programmation et l'informatique qui étaient entrés dans les programmes de 2015, parfois même de 2008. Par exemple, au cycle 4, le tableau 1 analyse la présence et l'évolution, dans les programmes de mathématiques, des termes : informatique, algorithmique / algorithmes, codage / coder / code, programmation / programmer, logiciel, robot et ordinateur.

Ces exemples montrent la complexité pour un enseignant de se repérer dans les prescriptions qui manquent de précisions et laissent même entrevoir des contradictions.

Au final, on peut se demander s'il y a une réelle volonté des prescripteurs de faire entrer les élèves dans la discipline informatique ou si quelques termes sont simplement éparpillés pour soutenir une volonté de former les citoyens aux technologies numériques.

## Variété et types de contenus informatiques

Les apports de travaux ancrés en informatique permettent de catégoriser les différents contenus d'enseignement relevant de l'informatique. Bruillard, à la suite des travaux de Mirabail (1990) et Dowek (2012) propose une catégorisation selon trois attracteurs : algorithmes, matériels et réseaux, activités humaines (Bruillard, 2009a).

Ce travail épistémologique sur l'informatique nous a conduits à construire une typologie de trois catégories de contenus possibles : les contenus peuvent concerner l'apprentissage de l'algorithmique ; ils peuvent concerner l'apprentissage du fonctionnement des technologies ; enfin, ils peuvent renvoyer à l'apprentissage de l'utilisation des outils informatisés. Ces catégories de contenus se retrouvent principalement dans les programmes ainsi que dans les manuels.

### Apprentissage du fonctionnement des technologies

L'apprentissage du fonctionnement des technologies consiste à enseigner aux élèves le fonctionnement des machines, mais aussi à enseigner l'historique des technologies et les enjeux du développement de ces technologies. À titre d'exemple, les programmes (MEN, 2015b) prévoient que les élèves soient capables de « décrire l'architecture simple d'un dispositif informatique » à la fin du cycle 2, ou encore de « comprendre le fonctionnement d'un réseau informatique » à l'issue du cycle 4.

Dans les ouvrages, les élèves sont par exemple amenés à apprendre comment relier des périphériques à l'unité centrale (manuel 5) ou encore à découvrir l'importance de l'informatique et de la notion

3 Relie les éléments à l'unité centrale en faisant les bons branchements.



FIGURE 2 – Exemple d'exercice relatif à l'apprentissage du fonctionnement des technologies : relier les périphériques à l'unité centrale (manuel 5 : Cohen, A., et Marcialis, J. (2018). *Comprendre les outils numériques et programmer*). © Éditions Hatier, 2020.

de boucle dans les chaînes robotisées de montage de voitures (manuel 2).

## Apprentissage de l'algorithmique

Dans les programmes d'enseignement (MEN, 2015b), il s'agit de « coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran » au cycle 2, de « concevoir et programmer des applications informatiques pour des appareils nomades » et d'« écrire, mettre au point et exécuter un programme » au cycle 4.

L'apprentissage de l'algorithmique, de la programmation, des langages, etc. consiste, par exemple, à amener l'enfant à écrire des séquences d'action (manuel 1), à lire un message codé avec des flèches afin de tracer sur un damier le parcours d'un robot virtuel (manuel 3), ou à coder avec Scratch (manuel 9).

## Apprentissage de l'utilisation des outils informatisés

L'apprentissage de l'utilisation des outils informatisés vise un usage raisonné des outils informatiques, des logiciels, des moteurs de recherche, etc. Ce type de contenus se retrouve dans les programmes d'enseignement. Au cycle 2, les enfants doivent notamment « avoir acquis une familiarisation suffisante avec le traitement de texte et en faire un usage rationnel » (MEN, 2015b). À l'issue du cycle 3 les enfants doivent savoir « écrire avec un clavier rapidement et efficacement » (*ibid.*), et ils doivent aussi faire « usage de logiciels usuels » (*ibid.*).

Au sein des manuels, il peut s'agir de s'approprier un clavier, un logiciel de traitement de texte via des exercices ou d'apprendre à utiliser un moteur de recherche ou d'envoyer un courriel (manuel 5).

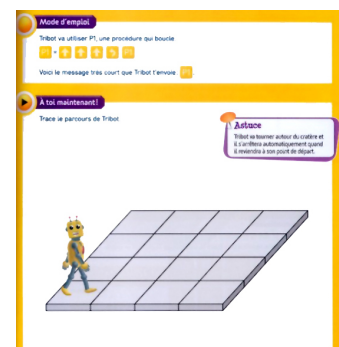


FIGURE 3 – Exemple d'exercice relatif à l'algorithmique : l'enfant doit tracer sur le damier, le parcours du robot défini par les flèches (manuel 3 : Croq, A., Farnet, P., Payet, N., et Tranny, G. (2015b). *Mon cahier pour apprendre à programmer*). © Éditions Bordas, 2015.



**1** Regarde ton clavier. Trouve les touches qui ont un repère en relief.

Les deux touches qui ont un repère en relief permettent de se situer tactilement sur le clavier.

**2** Barre les suites de lettres ou de signes qui ne sont pas sur ton clavier.

AZERTYUIOP	AZERPOIUYT	1234567890*+
NBVCXW	QSDFGHJKLM	APZOEIRUTY
&é"(-è_çà)=	WXCVBN	MLKJHGFDSQ

**3** Complète les lignes du clavier avec les lettres ou les signes qui manquent.




FIGURE 4 – Extrait d'une page d'exercices relatifs à l'apprentissage de l'utilisation des outils informatisés (manuel 5 : Cohen, A., et Marcialis, J. (2018). *Comprendre les outils numériques et programmer*). © Éditions Hatier, 2020.

## Distribution des contenus dans des manuels scolaires entre fonctionnement, algorithmique et utilisation

Dans un échantillon de dix manuels destinés aux enfants de l'école primaire, comment se répartissent quantitativement les contenus ? L'informatique en tant que technologie est très sous-représentée (6 %) dans les ouvrages analysés tandis que l'algorithmique est majoritaire (59 %) (Vandevelde, *et al.*, 2022).

Nous avons pu montrer (*ibid.*) une évolution de la catégorie de contenu selon le cycle. Parmi les ouvrages analysés, ceux à destination des cycles 1, 2 et 3 se concentrent sur les questions d'algorithmique. La question de l'utilisation n'apparaît qu'à partir du cycle 3. Les contenus liés à la dimension technologique de l'informatique se concentrent sur les cycles 2 et 3.

Qu'en est-il des types de contenus informatiques au sein des programmes d'enseignement ? Une logique similaire se dégage-t-elle ? Pour cela, nous avons comptabilisé tout ce qui nous apparaissait, dans les programmes scolaires, comme un contenu au sens défini plus haut, puis nous avons catégorisé les 53 contenus identifiés selon les trois catégories (l'imprécision de certaines formulations issues des programmes nous a poussés à classer deux contenus au sein de deux catégories différentes, ce qui fait 55 contenus catégorisés).

Nous commençons par croiser, au sein des programmes, les quatre cycles d'enseignement et les trois types de contenus informatiques identifiés. Les programmes d'enseignement portent surtout sur

les questions d'utilisation de l'informatique (49 %) et sur l'algorithmique (33 %), et la quantité de contenus abordés croît au fur et à mesure de l'avancée dans les cycles d'enseignement, passant de deux contenus au cycle 1, à 35 contenus au cycle 4.

Si l'on regarde ici aussi l'évolution en fonction des cycles, c'est surtout le contenu relatif à l'apprentissage de l'utilisation des technologies qui est présent pour les cycles 1 à 3 et le type de contenu relatif à l'apprentissage de l'algorithmique qui est présent pour le cycle 4. Nous remarquons également une véritable évolution au cours des cycles : il y a une interaction statistiquement significative entre le cycle et la catégorie de contenu ( $\chi = 9,02$ ,  $dl = 6$ ,  $\rho = 0,001$ ). Premièrement sur l'apprentissage de l'utilisation des outils informatisés qui commence au cycle 1 (découverte de divers outils), s'affine au cycle 2 (dactylographie), continue au cycle 3 (utilisation de logiciels usuels), et se termine au cycle 4 (utilisation des outils en réseaux). Deuxièmement, l'apprentissage de l'algorithmique est absent au cycle 1, démarre dès le cycle 2 (codage de déplacements) puis se perfectionne au cycle 3 (instauration de notions spécifiques) pour se parachever au cycle 4 (application des concepts étudiés).

Ces résultats de l'analyse des programmes diffèrent de ceux établis pour les ouvrages scolaires. D'après les programmes d'enseignement, les cycles 1 à 3 (qui correspondent à l'école maternelle et élémentaire et à la première année du collège français) se concentrent sur l'utilisation des technologies, la question de l'algorithmique étant surtout à travailler au cycle 4. À l'inverse, les manuels pour les cycles 1 à 3 se concentrent sur les questions d'algorithmique, alors que dans les ouvrages du cycle 4 c'est surtout l'utilisation des technologies qui est travaillée.

Ces différences peuvent notamment s'expliquer par le fait que les programmes, s'ils listent les compétences à acquérir à l'issue de chacun des cycles, ne donnent pas d'information quant à la durée à consacrer à une compétence par rapport à une autre. Dans notre analyse, chaque compétence du programme scolaire a été comptabilisée comme un contenu unique, mais il est possible que la traduction dans les livres de cette compétence représente plusieurs « éléments textuels » (pages de textes ou exercices).

Cycle	Algorithmique	Utilisation	Technologie	Total
Cycle 1	0	2	0	2
Cycle 2	1	4	1	6
Cycle 3	1	8	3	12
Cycle 4	16	13	6	35
Total	18	27	10	55

**TABLEAU 2** – Tableau croisé : les catégories de contenus en fonction des cycles, au sein des programmes d'enseignement.

## Fonctionnement pédagogique-didactique

Sous le terme « fonctionnement pédagogique-didactique », Reuter (2014) regroupe des indicateurs portant sur les contenus, les dispositifs, les activités d'enseignement, d'apprentissage et d'évaluation. Les textes institutionnels ne rentrent pas dans le détail du fonctionnement. C'est dans les programmes que des précisions sont apportées quant à la manière de travailler les contenus relevant de l'informatique. Dans les programmes de 2015 (MEN, 2015a ; 2015b), ces précisions prennent la forme « d'exemples de situations, d'activités et de ressources pour l'élève » proposés aux enseignants afin que les élèves acquièrent les « connaissances et compétences » (les contenus) requises.

Pour chaque contenu informatique identifié, un exemple d'activité n'est pas systématiquement donné. Ainsi, au cycle 3, les élèves doivent « apprendre à utiliser les outils numériques qui peuvent conduire à des réalisations collectives » ou encore apprendre à « utiliser des logiciels d'acquisition de données, de simulation et des bases de données » sans qu'aucun exemple de situation, d'activité ou de ressource pour l'élève ne soit mentionné.

Que ce soit au niveau des contenus ou des exemples d'activités et de ressources, les outils à utiliser pour les enseignements sont mentionnés d'une manière assez peu précise. Ainsi, les enseignants de cycle 2 sont invités à initier les élèves « à l'usage d'un logiciel permettant de représenter les solides » (MEN, 2015b). Les élèves de cycle 3 doivent savoir « utiliser des outils numériques » (MEN, 2015b), mais aucune précision n'est donnée quant à la nature de ces outils numériques. Les quelques rares outils numériques clairement explicités dans les programmes d'enseignement sont les ordinateurs (aux cycles 1 à 4), les claviers, les logiciels de traitement de textes et les robots.

Une autre manière de comprendre le fonctionnement pédagogique-didactique consiste à interroger les modalités de travail des élèves. Notamment, les élèves doivent-ils lire (afin de comprendre et/ou retenir) ou réaliser des activités ? Puisque nous avons constitué nos éléments textuels selon qu'il s'agissait de pages de cours

TABLEAU 3 – Extrait des programmes d'enseignement du cycle 2 (MEN, 2015).

<b>Commencer à s'approprier un environnement numérique</b>	
Décrire l'architecture simple d'un dispositif informatique.	Observer les connexions entre les différents matériels.
Avoir acquis une familiarisation suffisante avec le traitement de texte et en faire un usage rationnel (en lien avec le français)	Familiarisation progressive par la pratique, usage du correcteur orthographique. Mise en page, mise en forme de paragraphes, supprimer, déplacer, dupliquer. Saisie, traitement, sauvegarde, restitution.

ou d'exercices, un premier critère très simple est de regarder la répartition entre ces deux modalités. Cet indicateur très frustré est néanmoins surprenant. Sur l'ensemble des ouvrages, il y a plus de cours que d'exercices (62 % de cours et 38 % d'exercices). En outre, sur les dix manuels analysés, six contiennent plus de pages de cours que d'exercices. Dans la mesure où il est généralement admis que l'informatique doit être pratiquée pour être apprise (Fluckiger, 2019a), nous nous attendions à trouver davantage d'exercices que de pages de cours. Nous pouvons donc dire que dans la conception qu'en ont les rédacteurs des ouvrages (au moins de six sur dix), l'informatique se pratique, certes (les élèves sont amenés à faire de l'informatique), mais elle s'explique ou se raconte beaucoup (les élèves sont amenés à se documenter sur l'informatique).

Nous avons également pu montrer (Vandeveldé *et al.*, 2022) qu'il existe une relation entre la modalité page de cours ou exercice et le type de contenu informatique abordé ( $\chi = 73,16$ ,  $df = 2$ ,  $p = 0,001$ ) : l'apprentissage de l'utilisation des technologies passe par des exercices plus que par des textes, tandis que l'apprentissage de l'algorithmique passe surtout par des textes. Il s'agit d'un résultat surprenant. On aurait pu faire l'hypothèse que les questions d'algorithmique et de programmation, nécessitant une certaine pratique de la part de l'élève, soient davantage abordées au travers d'exercices, et que les questions d'utilisations seraient plutôt abordées dans les cours. Or, les résultats de l'analyse des dix manuels sélectionnés, sans que notre choix subjectif n'entre en jeu, ne vont pas dans ce sens.

## Modes de références des contenus aux espaces théoriques

Reuter désigne par le terme de « relations aux espaces théoriques » les questions relatives à la construction des savoirs scolaires (Reuter, 2014). À la suite de Martinand (1986), nous considérons que les contenus scolaires ne sont pas issus uniquement de savoirs académiques, mais peuvent être construits en référence à des pratiques de production, d'ingénierie, voire des pratiques quotidiennes dans différentes sphères sociales. C'est pourquoi nous cherchons à déterminer dans quelle mesure les contenus sont référés à des savoirs académiques ou à diverses pratiques sociales.

Les textes institutionnels se revendiquent volontiers de la nécessité d'adapter l'école et les enseignements à l'évolution de la société et sa numérisation rapide. Mais comment une telle recommandation est-elle mise en œuvre et comment se traduit-elle ?

Dans les programmes, les exemples d'activités et de ressources concernant l'apprentissage de contenus informatiques font largement référence à des situations « technologisées », dans le sens où l'utilisation d'outils numériques, même si la nature de ceux-ci n'est que peu précisée, est quasiment exclusive au sein des exemples de situations, d'activités ou de ressources pour l'élève, proposés aux enseignants. Néanmoins, le fait que les contenus informatiques soient travaillés au sein d'autres enseignements plus clairement identifiés tels que les mathématiques, le français ou autre, tend vers une double contextualisation des enseignements : une première contextualisation venant de l'enseignement dans lequel l'apprentissage de l'informatique s'inscrit, et une seconde contextualisation venant de la mobilisation d'outils technologiques.

De ce point de vue, il est frappant de constater que dans les manuels, les situations d'apprentissage proposées ne font pas toujours référence à des situations « technologisées ». Une telle présentation doit beaucoup à l'idée qu'il faudrait que les exercices scolaires soient « proches de la vie réelle », qui imprègne par exemple les grandes évaluations comme le Programme international pour le suivi des acquis des élèves (PISA) (Bart et Fluckiger, 2015).

Ainsi, dans l'ouvrage de cycle 1, la narration et les exercices proposés font référence à des situations supposées familières aux enfants et ne semblent *a priori* pas faire référence aux ordinateurs ou à l'informatique. La notion de boucle est introduite via la construction d'une échelle, les motifs répétés sont travaillés sur des séquences de papier peint, etc.

Pourtant un lecteur attentif et plus connaisseur peut déceler des références discrètes à l'informatique et à la culture numérique, à travers le choix des personnages et des représentations graphiques qui illustrent l'ouvrage (manuel 1, figure 12) : le personnage du Léopard des neiges, allégorie de la mise à jour du système

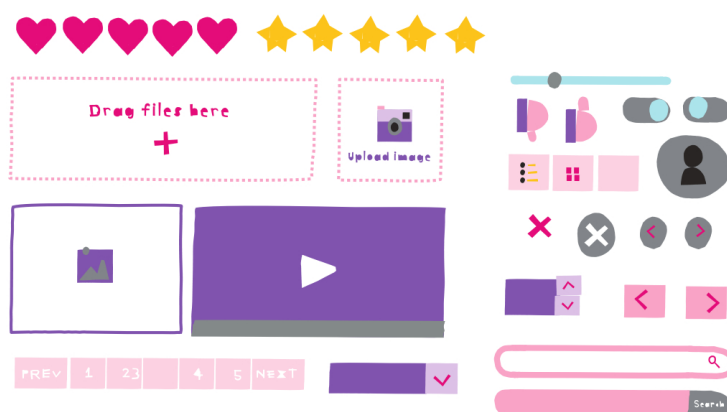


FIGURE 5 – Illustration issue du projet Hello Ruby (<https://www.helloruby.com>), à l'origine du manuel 1. CC BY-NC-SA.

d'exploitation Apple; des robots évoquant le logo du système d'exploitation Android; un motif de tapisserie ressemblant au logo de la marque Apple; une illustration faisant penser au circuit imprimé et processeur d'une carte mère, etc. (figure 5).

À partir du cycle 2 puis dans les manuels des cycles 3 et 4, la présence du monde informatique au sens large passe également par des choix graphiques. Au cycle 3, un manuel fait se déplacer des personnages pixélisés, typiques de l'esthétique informatique des années 1980, évoquant un ancien jeu vidéo. Au cycle 4 l'univers graphique est moins présent, l'idée de machine n'est plus représentée que par le robot mBot, robot réel que les enfants peuvent être amenés à programmer en classe.

## Visées ou finalités assignées aux contenus informatiques à l'école primaire française

Une explicitation des enjeux et finalités d'un enseignement de l'informatique dès l'école primaire est surtout le fait des textes institutionnels, plus rarement des programmes. Ainsi, le CNUM (texte 2) précise qu'enseigner l'informatique « est une réponse à l'attente sociale d'une politique de l'égalité : permettre à tous les élèves d'avoir une « clé » pour comprendre le monde numérique, participer à la vie sociale et se préparer à de nouveaux mondes professionnels ». Il s'agit alors de « mieux comprendre le monde numérique qui nous entoure » ou, comme le dit Studer (texte 1), de « donner à nos enfants les clefs de leur vie numérique ». La finalité est ici politique et citoyenne, les enfants étant appelés à devenir les citoyens de demain : pour les auteurs de ces textes, comprendre le monde numérique est une condition pour « être pleinement un citoyen actif dans la société » (texte 2).

Cette visée d'une culture numérique générale, d'une formation à une culture scientifique et technique partagée passe cependant, pour les institutions éducatives, par une construction de « compétences dans le domaine du numérique » (texte 4). Ces compétences sont censées être construites par des expériences concrètes que les élèves peuvent vivre et poursuivre, dans un cadre scolaire ou hors temps scolaire » (*ibid.*). En effet, pour les rédacteurs, il semble y avoir une continuité entre les apprentissages extrascolaires concernant le numérique et les apprentissages scolaires : les élèves sont censés « réinvestir certains apprentissages informels, acquis en dehors de l'école » (texte 1), ce qui était déjà la logique présidant au certificat de compétences B2i (Fluckiger et Bart, 2012).

## Conclusion

Ce chapitre visait à présenter brièvement comment les contenus informatiques se présentent aux acteurs de l'enseignement (enseignants, élèves, parents) sous une forme de textes et de discours : quels types de contenus, organisation, désignation, objectifs affichés dans la scolarité des élèves.

Ce sont les réflexions sur les disciplines scolaires qui nous ont permis de dégager les grandes lignes des enseignements prescrits. Cela n'est pas si paradoxal : en effet, ces contenus ne se présentent pas sous la forme d'une discipline solidement établie, dotée d'un corps enseignant spécifique. Mais cela est rarement, sinon jamais le cas à l'école primaire. Surtout, une discipline est bien une construction sociale, dans tous les sens du terme : non seulement elle a une histoire, inscrite dans les évolutions sociales et culturelles de la période (du même ordre que celles qui ont conduit à la quasi-disparition des disciplines autrefois reines qu'étaient le grec et le latin), mais aussi et surtout les disciplines sont (re)construites par les acteurs, qui se saisissent des discours institutionnels pour se faire une idée de ces enseignements, de leur organisation, leurs modalités ou leurs finalités.

Cette analyse d'un corpus textuel encadrant et guidant les enseignements ne dit évidemment rien de la façon dont ces textes seront appréhendés et mis en œuvre par les enseignants, ni comment les élèves construiront des représentations quant à aux savoirs en jeu dans les activités scolaires qui leur sont proposées autour de l'informatique ou de la robotique scolaire. C'est ce que proposent les chapitres suivants.

## Recommandations

Concernant les désignations des contenus informatiques, les confusions constatées dans la manière de désigner les contenus informatiques nous conduisent à recommander de bien distinguer dans les manuels et les programmes d'enseignement ce qui relève de l'informatique de ce qui relève des usages du numérique.

Suite à notre analyse des différents types de contenus présents dans les programmes et les manuels, nous préconisons que les différentes dimensions d'une culture informatique soient prises en compte lors des enseignements (concernant la science informatique et l'algorithmique, concernant la technologie des ordinateurs, des tablettes numériques ou des réseaux, concernant l'usage des outils et logiciels courants). Les trois dimensions peuvent faire l'objet de cours comme d'exercices.

Enfin, les liens entre les savoirs issus de la science informatique et l'environnement largement numérisé, la technologie numérique, dans laquelle évoluent les enfants ne sont pas toujours clairement mis en avant. Les exemples de situations invoquant l'algorithmique sont souvent tirés de situations éloignées de la technologie numérique (construire une échelle, faire un gâteau. . .), ce qui ne permet pas aux enfants de bien comprendre que c'est ainsi que fonctionnent les outils numériques. Il conviendrait de penser davantage les enseignements informatiques, dans une scolarité obligatoire, comme un moyen de donner des clés de compréhension du fonctionnement des technologies numériques, en les inscrivant dans une approche sociétale.

## Références

Amadiou, F., et Tricot, A. (2014). *Apprendre avec le numérique. Mythes et réalités*, Retz. <https://doi.org/10.14375/NP.9782725633206>

Baron, G.-L., et Boulc'h, L. (2011). Les technologies de l'information et de la communication à l'école primaire. État de question en 2011, *Revue de l'enseignement public et informatique (EPI)*. <https://www.epi.asso.fr/revue/articles/a0910c.htm>

Baron, G.-L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique. *Revue Française de Pédagogie*, vol. 195, n° 2 p. 51-62. <https://doi.org/10.4000/rfp.5032>

Bart, D., et Fluckiger, C. (2015). « Évaluation, fabrication des contenus et disciplines d'enseignement », dans B. Daunay, C. Fluckiger, et R. Hassan, *Les Contenus d'enseignement et d'apprentissage. Approches didactiques*, Presses universitaires de Bordeaux, p. 91-102. <https://doi.org/10.4000/books.pub.38357>

Bruillard, É. (2009a). « Place de l'informatique dans l'enseignement secondaire, réflexions introductives », dans G.-L. Baron, É. Bruillard, et L.-O. Pochon (dir.), *Informatique et progiciels en éducation et en formation*, p. 21-27, INRP.



- Bruillard, É. (2014). *Une voie pour penser et construire une formation à l'informatique pour les élèves de l'école primaire ?*, Université Paris Cité. <https://hal.science/hal-03948939>
- Chevallard, Y. (1985). *La transposition didactique. Du savoir savant au savoir enseigné*. Le Pensée Sauvage.
- Daunay, B., Fluckiger, C., et Hassan, R. (2015). *Les contenus d'enseignement et d'apprentissage. Approches didactiques*. Presses universitaires de Bordeaux. <https://doi.org/10.4000/books.pub.38242>
- Delcambre, I. (2013). « Contenus d'enseignement et d'apprentissage », dans Y. Reuter (dir.), *Dictionnaire des concepts fondamentaux des didactiques*, 3<sup>e</sup> éd., De Boeck, p. 43-48. <https://doi.org/10.3917/dbu.reute.2013.01.0043>
- Dowek, G. (2012). « Les quatre concepts de l'informatique », dans G.-L. Baron, É. Bruillard, et V. Komis, *Actes du quatrième colloque international DIDAPRO 4—Dida&Stic*, New Technologies Editions, p. 21-29.
- Fluckiger, C. (2011). La didactique de l'informatique et les constructions sociales de la figure des jeunes utilisateurs, *Recherches en Didactiques*, vol. 11, p. 67-84. <https://doi.org/10.3917/rdid.011.0067>
- Fluckiger, C. (2019a). *Une approche didactique de l'informatique scolaire*, Presses universitaires de Rennes.
- Fluckiger, C. (2019b). Numérique en formation : des mythes aux approches critiques, *Education permanente*, vol. 219, p. 19-30. <https://doi.org/10.3917/edpe.219.0019>
- Fluckiger, C., et Bart, D. (2012). L'introduction du B2i à l'école primaire : évaluer des compétences hors d'une discipline d'enseignement ?, *Questions Vives*, vol. 7, n° 17, p. 71-87. <https://doi.org/10.4000/questionsvives.1006>
- Fluckiger, C., et Reuter, Y. (2014). Les contenus « informatiques » et leur(s) reconstruction(s) par des élèves de CM2. Étude didactique, *Recherches en éducation*, vol. 18, p. 64-78. <https://doi.org/10.4000/ree.8510>
- Harlé, I. (2016). Analyse de reconfigurations disciplinaires : les apports de la didactique à la sociologie, *Spirale*, vol. 58, p. 23-34. <https://doi.org/10.3917/spir.058.0023>
- Livingstone, S. (2012). Critical reflections on the benefits of ICT in education, *Oxford review of education*, vol. 1, n° 38, p. 9-24. <https://doi.org/10.1080/03054985.2011.577938>
- Martinand, J.-L. (1986). *Connaître et transformer la matière*, Peter Lang.
- Mirabail, M. (1990). La culture informatique, *ASTER*, vol. 11, p. 11-28. <https://doi.org/10.4267/2042/9117>

Reuter, Y. (2004). Analyser la discipline : quelques propositions, *Actes du 9<sup>e</sup> colloque de l'association internationale pour la recherche en didactique du Français*, vol. 35, p. 5-12. <https://doi.org/10.3406/airdf.2004.1611>

Reuter, Y. (2007b). « Discipline scolaire », dans Y. Reuter, C. Cohen-Azria, B. Daunay, I. Delcambre, et D. Lahanier-Reuter, *Dictionnaire des concepts fondamentaux des didactiques*, p. 85-89.

Reuter, Y. (2014). Construire la catégorie de discipline scolaire en didactique(s). *Linguarum Arena*, vol. 5, p. 79-95.

Reuter, Y., et Lahanier-Reuter, D. (2007). L'analyse de la discipline : quelques problèmes pour la recherche en didactique, *9<sup>e</sup> colloque de l'association internationale pour la recherche en didactique du Français*. <https://doi.org/10.3406/airdf.2007.1729>

Vandevelde, I., Fluckiger, C., et Nogry, S. (2022). Resources and textbooks for computer science education in French primary schools, *IARTEM E-Journal*, vol. 14, n° 1, p. 1-20. <https://doi.org/10.21344/iartem.v14i1.954>

Wing, J. M. (2006). Computational thinking, *Communications of the ACM*, vol. 49, n° 3, p. 33-35. <https://doi.org/10.1145/1118178.1118215>

## Annexes

### Manuels analysés

#### Cycle 1 :

Manuel 1 : Liukas, L. (2016). *Hello Ruby!*, Glénat jeunesse.

#### Cycle 2 :

Manuel 2 : Croq, A., Farnet, P., Payet, N., et Trannoy, G. (2015a). *J'apprends à programmer tout seul!* Bordas.

Manuel 3 : Croq, A., Farnet, P., Payet, N., et Trannoy, G. (2015b). *Mon cahier pour apprendre à programmer*, Bordas.

Manuel 4 : Lyons, H., et Tweedale, E. (2017). *Mon atelier code. Tout pour faire ses premiers pas en programmation informatique*, Fleurus.

#### Cycle 3 :

Manuel 5 : Cohen, A., et Marcialis, J. (2018). *Comprendre les outils numériques et programmer*, Hatier.

Manuel 6 : Morgan, N. (2017). *Javascript pour les kids*, Eyrolles.

Manuel 7 : Vorderman, C., Woodcock, J., et McManus, S. (2017). *À vos marques, prêts ? Codez !* Larousse.

#### Cycle 4 :

Manuel 8 : Anguenot, G., Launay, J., Corne, R., Vogt, O., et Sauzeau, D. (2016). *Cahier d'algorithmique et de programmation*, Delagrave.

Manuel 9 : Bassette, T. (2017). *1, 2, 3, je code avec Scratch*, Larousse.

Manuel 10 : Plumel, D. (2017). *1, 2, 3, je construis avec Minecraft*, Larousse.

### Textes institutionnels

Texte 1 IGEN (2017). *Repenser la forme scolaire à l'heure du numérique. Vers de nouvelles manières d'apprendre et d'enseigner*, Rapport à monsieur le ministre de l'Éducation nationale, Inspection générale de l'Éducation nationale [https://cache.media.education.gouv.fr/file/2017/55/1/IGEN-Rapport-2017-056-Repenser-forme-scolaire-numerique-nouvelles-manieres-apprendre-enseigner\\_849551.pdf](https://cache.media.education.gouv.fr/file/2017/55/1/IGEN-Rapport-2017-056-Repenser-forme-scolaire-numerique-nouvelles-manieres-apprendre-enseigner_849551.pdf)

Texte 2 CNNum (2014). *Jules Ferry 3.0, Bâtir une école créative et juste dans un monde numérique*, Conseil national du numérique. [https://cnnumerique.fr/files/2017-10/Rapport\\_CNNum\\_Education\\_oct14.pdf](https://cnnumerique.fr/files/2017-10/Rapport_CNNum_Education_oct14.pdf)

Texte 5 CACE (2018). *Rapport d'information*, Commission des affaires culturelles et de l'éducation. <http://www.assemblee-nationale.fr/15/rap-info/i1296.asp>

Texte 3 Institut Montaigne (2016). *Le numérique pour réussir dès l'école primaire*, Rapport de mars 2016 de l'institut Montaigne [https://www.institutmontaigne.org/ressources/pdfs/publications/institut\\_montaigne\\_le\\_numerique\\_pour\\_reussir\\_des\\_l\\_ecole\\_primaire.pdf](https://www.institutmontaigne.org/ressources/pdfs/publications/institut_montaigne_le_numerique_pour_reussir_des_l_ecole_primaire.pdf)

Texte 4 MEN (2018). *Le numérique au service de l'école de la confiance*, Ministère de l'Éducation nationale. <https://www.education.gouv.fr/le-numerique-au-service-de-l-ecole-de-la-confiance-3212>

## Corpus de programmes

Prog. 1 MEN. (2015a). Programme d'enseignement de l'école maternelle.

Prog 2-3-4 MEN. (2015b). Programmes d'enseignement du cycle des apprentissages fondamentaux (cycle 2), du cycle de consolidation (cycle 3) et du cycle des approfondissements (cycle 4).

S4C MEN. (2015c). Socle commun de connaissances, de compétences et de culture : Décret n°2015-372 du 31-3-2015.

Prog.2020 MEN (2020). Programmes d'enseignements Cycles 2, 3 et 4, Modifications. <https://www.education.gouv.fr/bo/20/Hebd031/MENE2018714A.htm>

Pour citer ce chapitre :

Vandevelde, Isabelle, Fluckiger, Cédric, Grugier, Olivier, et Haspekian, Mariam (2024). « Quel enseignement de l'informatique dans la scolarité obligatoire en France? Analyse des programmes et manuels », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 27-49. <https://doi.org/10.53480/2024iecare02p/>



# Pratiques envisagées des enseignants pour un enseignement de l'informatique à l'école primaire

## 3

Cédric FLUCKIGER<sup>1</sup>  
Olivier GRUGIER<sup>2</sup>  
Mariam HASPEKIAN<sup>2</sup>

1. Université de Lille, ULR 4354, CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, 59000 Lille, France

2. Université Paris Cité, Laboratoire EDA, 75006 Paris, France

« Je suis une scientifique de base [...] l'informatique je maîtrise plutôt pas mal ». C'est ainsi qu'Isabelle, une enseignante d'école primaire française, justifiait le fait qu'elle était la seule dans son école à avoir mis en place un enseignement de l'informatique. Les autres enseignants de l'école lui délèguent d'ailleurs cette responsabilité : l'équipe éducative a mis en place un temps de décroisement pour que toutes les classes accomplissent, sur une période, un temps d'informatique avec Isabelle. Cet exemple illustre le fait que même si des contenus sont prescrits pour un enseignement de l'informatique, parfois même avec l'incitation des collectivités territoriales qui gèrent les équipements dans les écoles, utilisant ce levier pour promouvoir une politique locale, les mettre en œuvre dans une classe est considéré par les enseignants comme un choix. Ce choix est guidé par la légitimité et l'importance accordées à ces contenus et à leurs visées, mais aussi par les ressources pédagogiques comme les activités proposées dans les manuels, les modes de travail pédagogique-didactiques possibles, etc.

Ce chapitre propose d'explorer les choix et intentions des enseignants quant à la mise en place d'un enseignement de l'informatique. Pour cela, les pratiques déclarées des enseignants relatives à un enseignement de l'informatique à l'école sont analysées afin de percevoir et de comprendre les choix des contenus envisagés ou enseignés dans les classes, ainsi que les modalités prévues pour leur mise en œuvre, notamment les choix des artefacts envisagés.

Dans les programmes prescrits pour les trois premiers cycles de la scolarité française, les contenus que nous qualifions d'informatique, ne sont pas regroupés derrière un intitulé clairement défini comme tel. Dans le chapitre 2 « Quel enseignement de l'informatique dans la scolarité obligatoire en France ? Analyse des programmes et manuels » du présent ouvrage, les auteurs montrent que les savoirs en informatique sont disséminés, pouvant

rendre la mise en œuvre difficile pour des enseignants généralistes et non-spécialistes de ce champ disciplinaire. Finalement, quels sont les choix didactiques envisagés par les enseignants des écoles pour mettre en œuvre le curriculum formel concernant les savoirs en informatique ?

Ce chapitre commencera par présenter le cadre théorique utilisé puis le recueil des données pour ensuite discuter des résultats obtenus, notamment à propos des choix didactiques des enseignants et des explications qu'ils en donnent. En fin de chapitre, un modèle de registres d'activité est présenté, permettant ainsi de situer l'activité de choix d'artefact. Quelques recommandations concluent le chapitre en vue d'ouvrir des pistes favorisant les pratiques enseignantes en informatique.

## **Cadre théorique pour discuter et analyser les choix des enseignants**

Notre questionnement s'inscrit dans une perspective que l'on peut qualifier de subjectiviste en didactique : nous nous intéressons ici aux points de vue des sujets (les enseignants), à leur manière de vivre, construire et reconstruire les disciplines scolaires. En effet ces dernières, en tant que constructions sociales, sont à la fois des modalités d'organisation des contenus scolaires (Dauney et Reuter, 2013) mais aussi des espaces vécus subjectivement par les sujets didactiques (Reuter, 2013).

Pour analyser cette manière subjective dont les acteurs investissent les contenus informatiques prescrits, nous recourons ici à des outils théoriques qui s'inscrivent dans les champs de la didactique des disciplines, en particulier celle des mathématiques, celle de la technologie et celle de l'informatique.

Développé en didactique des mathématiques, le cadre de la double approche didactique et ergonomique de Robert et Rogalski (2002) vise à approcher les pratiques d'enseignement sous un angle à la fois didactique et ergonomique. Pour les comprendre, il questionne en effet les choix didactiques que font les enseignants, tout en tenant compte de contraintes liées au métier. L'enseignant est libre de ses choix cognitifs (tâches proposées aux élèves, progression didactique, etc.) et médiatifs (modalités concrètes associées à ces tâches et à cette progression). Il effectue ces choix en fonction de son bagage et sa sensibilité personnels (formation, expérience, représentation du métier, des disciplines à enseigner, de la façon dont les élèves apprennent, etc.). Ses marges de manœuvre sont cependant limitées par ce que lui impose l'institution (programmes, horaires. . .) et par le contexte social dans lequel il enseigne (établissement défavorisé, élèves non francophones, etc.).

Ces cinq composantes (cognitive, médiative, personnelle, institutionnelle et sociale) sont utilisées ici pour analyser ce que disent des enseignants ordinaires de ces programmes.

L'enseignement de l'informatique reste, par ailleurs, étroitement lié à l'utilisation d'artefacts, que nous définissons comme des objets ou des systèmes techniques ou symboliques ayant subi une transformation d'origine humaine, permettant de viser des contenus d'apprentissage. Le cadre de Combarous (1984), utilisé dans d'autres travaux de recherche (Grugier, 2021), avec les trois composantes de la technicité, permet d'effectuer une analyse à la fois didactique et technique des pratiques enseignantes à travers leurs choix d'artefacts. La technicité résulte de la réunion et de l'interaction permanente de trois composantes : la réflexion technique dans les choix de nature didactique, la présence de matériels pour l'apprentissage de l'informatique, et une composante sociale susceptible d'influencer et d'orienter les pratiques enseignantes.

Enfin, la didactique de l'informatique s'attache à déterminer quels peuvent être les contenus et les différents sens de ces apprentissages pour les élèves. Des travaux ont montré que l'informatique est « à la fois une science, mais aussi des usages, l'utilisation de cette science, et enfin un ensemble d'outils, c'est-à-dire une technologie » (Fluckiger, 2017) et que « ces trois aspects sont à enseigner » (*ibid.*). Cependant la question de la programmation est masquée aujourd'hui par les applications de l'informatique, comme rappelé dans le chapitre précédent.

Fluckiger (2017) dégage plusieurs caractéristiques propres à cet enseignement tel qu'il se présente actuellement en France, dont certaines se retrouvent, nous le verrons, dans les discours des enseignants : petite ancienneté de la discipline scolaire au regard de son ancienneté dans les universités, poids institutionnel réduit, notamment en termes d'inscription horaire dans les emplois du temps et de manque d'examen, absence d'une légitimité en soi au lieu d'une légitimité fondée sur des usages extrascolaires des élèves, cette absence allant de pair avec la question de l'autonomie et de l'ancillarité de l'informatique (discipline non autonome et toujours « au service de »), et image véhiculée dans la société associant l'informatique à des actions plus qu'à des savoirs.

## Méthodologie

Nous nous appuyons ici sur trois corpus de données visant à recueillir les choix des enseignants quant à la mise en œuvre des contenus informatiques prescrits.

D'une part, nous avons administré à plus de 600 enseignants expérimentés et enseignants stagiaires de quatre académies un



questionnaire construit selon les catégories de la technicité (Combarrous, 1984), explorant les dimensions cognitive, matérielle et sociale des choix qui président aux enseignements informatiques. Soixante-dix retours sont exploitables. D'autre part, nous avons interrogé 11 enseignants d'école primaire ayant des profils variés, désignés par l'inspectrice d'une circonscription, mettant en œuvre ou non ces enseignements, enthousiastes ou réfractaires à cet enseignement. Chaque entretien individuel et semi-directif avait une durée de 1 h. Ces entretiens ont été retranscrits et analysés avec les catégories issues des deux cadres théoriques mobilisés ici. On s'intéresse ainsi aux contraintes déclarées par les enseignants, aux apprentissages qu'ils visent ainsi qu'aux modalités mises en œuvre effectivement ou envisagées, en mettant en regard ces éléments avec des caractéristiques personnelles.

Le choix des contenus à enseigner étant, de notre point de vue, inséparable du choix des outils pour les mettre en œuvre, nous avons enfin mené une enquête, avec une prise de contact par mail et par téléphone, auprès des 11 principaux fournisseurs de matériel scolaire en France. En supposant que le choix d'un artefact tel qu'un robot programmable et son utilisation en classe vont orienter les contenus enseignés, nous avons cherché à identifier ceux qui sont les plus commercialisés et livrés dans les établissements scolaires.

Les deux parties suivantes présentent nos résultats. La première, mobilisant le cadre de la double approche, permet de comprendre dans quelle mesure les choix didactiques des enseignants sont ancrés dans un rapport personnel à l'informatique. La seconde examine la composante cognitive en s'intéressant plus précisément aux choix étroitement liés de contenus et d'artefacts des enseignants.

## **Pratiques envisagées des professeurs des écoles pour un enseignement d'informatique**

### **Contraintes déclarées des enseignants**

Les discours des enseignants, même lorsqu'ils sont favorables à un enseignement de l'informatique à l'école, font état de contraintes importantes.

Une première contrainte est liée au besoin de formation et d'échanges entre pairs, ressenti comme nécessaire pour un enseignement au statut scolaire aussi instable que l'informatique. Ainsi, Sabine, enseignante en cours moyen 2<sup>e</sup> année (CM2), justifie ses réticences à mettre en place cet enseignement par l'absence de formation :

Personnellement je n'ai eu qu'une petite formation de base à l'Institut universitaire de formation des maitres (IUFM) et puis après je n'ai plus rien eu. Au niveau des animations pédagogiques ou au niveau des propositions qu'on a eues après, rien. Il y a un réel manque [...], on nous demande de valider des compétences, d'être à l'ère du numérique mais on ne nous aide pas.

De même Manon (CM2) dit : « tout ce qui est programmation, il me faudrait une formation pour le travailler avec mes élèves ». Par ailleurs, le fait d'avoir suivi une formation ne suffit pas à mettre en œuvre, ni à se sentir à l'aise avec ces enseignements. Certains ont suivi une formation sans l'enseigner, d'autres enseignent sans avoir suivi de formation mais indiquent, comme Tom, s'être auto-formés : « Moi le codage je le fais avec ces petits robots mais c'est parce que j'ai appris à les utiliser sur mon temps personnel, toujours en auto-formation ».

Une seconde contrainte concerne l'équipement matériel. Les écoles sont souvent sous-dotées en matériel informatique (logiciels de programmation ou robots). Le manque de matériel et leur accès restreint (parfois à une plage unique fixée dans la semaine) conduisent les enseignants non seulement à ne pas utiliser de technologies lorsqu'ils le souhaitent, mais à ne pas même être en mesure de s'approprier les outils nécessaires à une activité en classe avec les élèves. C'est le cas d'Anne-Claire (CM1-CM2), qui raconte sa « connexion limitée dans l'école » :

Quand on veut faire de l'anglais on part toujours d'une chanson ou d'une petite vidéo pour faire parler les enfants mais la connexion elle bugge à chaque fois donc on doit regarder la vidéo ou écouter le chant sur mon téléphone ! Si j'étais mieux équipée, à force d'utilisation je deviendrais plus à l'aise avec tout ça.

Mais ce sont surtout des contraintes liées au fonctionnement institutionnel dont font état les enseignants, à l'instar de Sabine, en CM2 :

Dans toutes les matières on a vraiment des heures définies, et même avec ces heures définies qui nous paraissent énormes on a du mal à boucler parfois notre programme, on doit s'adapter aux élèves en difficulté. Donc là on va peut-être être un peu plus guidés et on va leur laisser moins de tâtonnement parce qu'on ne peut pas se permettre, sur la progression qu'on a fixée, d'accorder énormément de temps à ce tâtonnement.

Ainsi, l'accent mis par l'école française sur le lire-écrire-compter peut se faire au détriment d'autres types de contenus qui, surtout s'ils sont nouveaux, méconnus et parfois mal maîtrisés, pâtissent plus que d'autres de cette concurrence avec des contenus jugés plus fondamentaux. Sandrine, en grande section (GS), le dit nettement : « [l'informatique], ce n'est pas un de mes objectifs, je préfère qu'ils connaissent les lettres, les sons etc. ».

## Apprentissages incertains

Les contenus et les apprentissages sont parfois mal identifiés. Ainsi, les enseignants sont nombreux, lorsqu'on les interroge sur l'enseignement de l'informatique, à répondre sur la technologie éducative au service d'autres apprentissages. C'est le cas de Stéphanie, enseignante en CM1, qui, à la question « est-ce que vous avez déjà enseigné de l'informatique ? » répond :

Oui j'ai une appétence particulière pour le numérique et j'ai essayé depuis quelques années d'inclure dans mes pratiques enseignantes l'utilisation de l'outil numérique. Pas faire du numérique pour du numérique, [...] c'était vraiment le numérique comme outil si je jugeais que c'était une plus-value pour les apprentissages.

On peut s'interroger sur ce que représente pour Stéphanie l'informatique (puisque à une question portant sur l'enseignement de celle-ci, elle répond par ses appétences pour le numérique). Quoi qu'il en soit, elle semble porter une attention consciente aux finalités associées à cet enseignement. Les enseignants reprennent ici à leur compte les confusions relevées dans les programmes d'enseignement et les discours d'accompagnement. Stéphanie explique d'ailleurs que, pour elle, « mettre un élève devant un traitement de texte et lui dire d'écrire ou de recopier telle chose sans intention derrière », cela « enlève le sens de l'utilisation du numérique ». Pour cette enseignante, c'est bien l'absence de finalités à ce qu'elle nomme « du numérique pour du numérique » qui pose problème. Car la même enseignante ne dirait pas forcément « faire de l'histoire pour de l'histoire » par exemple... On retrouve ici des résultats mis en évidence en didactique de l'informatique (Fluckiger, 2017) : l'informatique n'a pas, dans les conceptions de cette enseignante représentative de beaucoup d'enseignants interrogés, le statut de contenus disciplinaires légitimes, mais rejoint plutôt des finalités extrascolaires. Elle l'exprime nettement : « pour moi le numérique n'est pas une discipline à part entière [...] l'outil numérique est au service de toutes les disciplines ». Cette tension entre autonomie et ancillarité des contenus des disciplines

scolaires fait partie des tensions classiquement repérées, notamment en didactique du français (Reuter, 2011). Presque tous les enseignants rencontrés expriment cette représentation :

Nathalie, en moyenne section (MS) : « On n'a pas des séances de numérique » ;

Lucas, en cours préparatoire (CP) : « Je ne fais pas une séance en me disant qu'aujourd'hui on va apprendre à utiliser le clavier. Ils apprennent à utiliser le clavier, à enregistrer un fichier etc. dans les séances de littérature, d'expression écrite etc. » ;

ou encore Tom (CM1) : « Ce n'est pas du temps informatique pur, c'est faire le diaporama et en même temps on acquiert des notions en informatique ».

Une autre difficulté rencontrée chez de nombreux enseignants est d'appréhender les enjeux de savoirs disciplinaires de l'informatique, au-delà des activités proposées lors de formations ou du matériel, même lorsqu'il est à disposition. Ainsi, Sandrine (GS), interrogée avec sa collègue Nathalie (MS), raconte qu'elle a réservé auprès de sa circonscription des robots BeeBot connus en formation. Elle souhaite travailler sur le quadrillage, « après on va apprendre à programmer la BeeBot pour le déplacement sur quadrillage ». Les objectifs d'apprentissage sont essentiellement liés aux objectifs de résolution de problèmes : « ça demande de la logique du point de vue des enfants, parce que la résolution de problème c'est dans les programmes, d'aller d'un endroit à un autre c'est résoudre un problème en trouvant le code, le chemin, donc ça rentre dans les programmes ». Mais à la question « [est-ce qu'il y a des] apprentissages qui seraient plus directement liés à des choses sur le code, les algorithmes ? », elles répondent :

Nathalie : Ça me rappelle vaguement quelque chose [rire]. Tu ne te souviens pas ? Les BeeBot ça montrait que tout était des choix de programmation, c'était vague, il y avait une histoire de . . .

Sandrine : Je ne vois pas.

Nathalie : Mais on nous en a parlé. [...] mais c'était il y a longtemps [rire], c'était il y a trois ans.

Sandrine : Je ne me souviens plus du tout.

Nathalie : Mais si, on avait fait notre petite vidéo en Stop motion.

Sandrine : Ah oui c'est vrai mais je ne me souviens pas de ça, de l'algorithme.

Nathalie : C'était il y a longtemps et c'était succinct ! Mais oui ça expliquait qu'une machine ça ne réfléchissait pas, que c'est nous qui dirigeons la machine, qui avons la main sur la machine. Oui, ce n'est déjà pas évident pour nous [rire].

Dans ces conditions, la plupart des enseignants rencontrés ne construisent pas les séquences informatiques ou numériques selon une programmation didactique de contenus disciplinaires liés à l'informatique. Beaucoup racontent des contenus non anticipés, qui « surgissent » au détour d'une question d'élève. Cette absence de programmation didactique, sans doute liée à une faible conscience des savoirs proprement informatiques à enseigner, est sans doute, pour les enseignants rencontrés, l'obstacle majeur à une intégration effective de contenus informatiques dans l'école primaire française.

### **Composante personnelle des enseignants**

Les données montrent qu'un des facteurs qui jouent de façon prépondérante dans la place accordée à cet enseignement de l'informatique est l'histoire personnelle. C'est le cas d'Isabelle, citée plus haut, qui explique ses pratiques d'enseignement de l'informatique en les rattachant à une qualité personnelle (être une « scientifique », au sens d'avoir fait des études de biologie avant de passer les concours de recrutement de l'Éducation nationale). Mais ce sont en réalité tous les enseignants qui expliquent leurs pratiques par une affinité plus ou moins grande avec la science informatique ou les usages des outils numériques (selon la confusion déjà relevée dans les discours enseignants).

Ainsi, pour Gaël (CP), ce qui explique qu'il fasse de l'informatique en classe tient au fait que « j'utilise dans mon quotidien [...] le traitement de texte [...] donc forcément ça me donne plus envie de leur enseigner ça ». Cette même idée peut se retrouver, par la négative, lorsque les enseignants disent qu'ils se limitent à donner quelques indications aux élèves, comme Manon (CM2), qui dit que « pour l'instant on va travailler tout ce qui est traitement de texte, recherche, mise en page » car c'est ce qu'elle utilise quotidiennement... et donc un outil pour lequel elle se sent légitime. Cette remise en cause de la légitimité de cet enseignement dans les programmes, par le besoin d'aller chercher d'autres légitimités comme si la légitimité institutionnelle ne suffisait pas, est une constante chez ces enseignants.

### Composante cognitive et médiative : quels apprenants ?

La question de la légitimité ressentie par les enseignants a à voir avec leur vision des apprenants et de leurs compétences. La perception des compétences des élèves est cependant ambivalente. Les enseignants affirment une certaine habileté des élèves, comme Sandrine (GS) qui dit que « parfois on est étonnées de certains enfants qui manipulent l'ordinateur limite mieux que nous. C'est pour ça que je ne fais pas de séances spécifiques au numérique pour apprendre à utiliser l'ordinateur ». Mais les mêmes peuvent aussi raconter des élèves démunis face à un clavier ou une souris. Cette ambivalence des discours nous semble liée au fait que la doxa sur les élèves compétents ne correspond pas à ce que voient les enseignants : ils constatent des usages quotidiens, mais plutôt de tablettes (comme Sandrine : « ils ont des tablettes parce que quand je leur dis de bouger la flèche, ils font ça sur l'écran [en faisant un geste de *swiper* avec le doigt] »), le clavier par exemple est souvent une découverte scolaire, et la compréhension des dimensions technologiques reste faible.

Au-delà des compétences prêtées aux élèves, beaucoup d'enseignants rencontrés voient l'exposition aux écrans numériques comme une menace. C'est le cas de Sandrine (GS) qui affirme qu'« il y en a beaucoup qui regardent les écrans avant d'arriver à l'école, j'ai une enfant [...] qui s'endort avec les écrans, elle a l'écran pendant deux ou trois heures le soir ». C'est aussi le sentiment de Manon (CM2) :

Même la prise des repas on fait manger d'abord les enfants après on les met devant la télé avant d'aller se coucher. Moi j'avais des élèves qui se couchaient devant la télé, ils s'endormaient devant la télé. Pour avoir eu des situations d'élèves où la télé c'est envahissant pour eux, des élèves qui ne sont plus concentrés, qui baillent facilement, qui ont les yeux rouges etc. [...] là je pense qu'il y a un travail à faire au niveau de l'école. Tout ce qui est éducation, exposition aux écrans.

Ainsi, un certain « rapport aux écrans » est, pour les enseignants rencontrés, un contenu scolaire en soi, justifié par la vision qu'ils ont de leurs élèves. Dans ce contexte, les enseignants voient comme une injonction contradictoire de devoir utiliser des outils numériques, des ordinateurs ou tablettes en classe et enseigner leur usage, alors même que les enfants sont décrits comme trop exposés aux écrans en dehors de l'école. Cela conduit à des limitations d'usages scolaires, comme Manon, qui explique : « Moi j'utilisais les tablettes pour garder des traces avec la fonction appareil photo, quand je faisais du langage j'enregistrais aussi avec les tablettes.

Mais ça se limitait à ça parce que limitation des écrans, éviter l'exposition aux écrans ».

C'est dans ce contexte, que les enseignants doivent choisir des contenus à enseigner ainsi que des artefacts (numériques, ou non, avec ou sans écran) pour porter ces apprentissages.

## Focus sur les choix de contenus et d'artefacts

### Quels contenus enseigner ?

La détermination des contenus qu'un enseignant peut ou non inclure dans l'activité de sa classe résulte, pour l'enseignant, de questionnements multiples. Déjà, les contenus doivent être perçus par les enseignants comme étant en lien avec les attentes du cycle : comme disait Sandrine précédemment, la résolution de problèmes « c'est dans les programmes ». Ainsi le contenu d'une séance de robotique, pour elle, est lié à la résolution de problèmes, ou à l'orientation dans l'espace décrite en mathématiques, non à des contenus disciplinaires informatiques (comme le concept de machine, d'instruction, de séquence d'instruction, d'algorithme, etc.).

Mais à ces préoccupations institutionnelles s'ajoutent d'autres, plus personnelles. Ici revient souvent le fait que ces contenus doivent en outre être en rupture avec les pratiques extrascolaires des élèves, précisément en raison de la perception chez les enseignants d'une forme de danger lié à ces usages. C'est ainsi que nous identifions dans les discours quelque chose qui est bien de l'ordre d'un contenu informatique, au sens selon lequel Delcambre (2013) propose de désigner non seulement des savoirs, mais aussi des savoir-faire, des rapports à, des valeurs. . . Beaucoup d'enseignants ont à cœur de faire acquérir aux élèves un rapport aux artefacts numériques, différent de celui qu'ils ont en dehors de l'école. Gaël justifie le fait de faire des séances de traitement de texte précisément parce que ce n'est pas un usage extrascolaire courant : « ils ont surtout un usage de jeux vidéo, de Youtube. [ . . . ] Justement le traitement de texte, ils n'en font pas à la maison donc ça me permet d'apporter ça ». Mais au-delà de la seule différence, c'est surtout une attitude, un « rapport à » différent que les enseignants entendent inculquer aux élèves : un rapport actif à la machine. Précisons que ce sont bien les didacticiens que nous sommes qui identifient là un contenu, les enseignants ne le formulent pas ainsi. Mais lorsque Nathalie (PM) explique que l'« on agit sur le problème du traitement de texte, parce que ça ils ne le font pas beaucoup à la maison, après les jeux ils en font assez facilement », c'est bien l'idée d'apprendre aux élèves un rapport autre qu'un rapport qu'elle estime passif. Sandrine l'exprime encore

plus nettement : « ce qu'ils font sur les tablettes [à la maison] c'est beaucoup de vidéos, ils ne sont pas forcément actifs devant leur tablette, ils sont passifs, ils reçoivent du contenu. Tandis que là ils doivent agir devant l'écran, c'est eux qui manipulent, ce sont eux qui font l'action ». On retrouve cette prégnance de l'action face aux savoirs évoquée plus haut en didactique de l'informatique ; ici, « manipuler », « agir devant l'écran » ne sont certes pas des savoirs disciplinaires de l'informatique, mais dans les discours des enseignants, ce sont d'importants contenus informatiques.

### Quels artefacts choisir ?

Comme plusieurs enseignants interviewés le soulignent, le matériel occupe une place importante dans les activités en informatique. C'est pourquoi il était nécessaire, pour comprendre les choix des contenus à enseigner par les enseignants, de nous pencher sur cette composante matérielle. Les programmes du cycle 2 mentionnent de « Programmer les déplacements d'un robot » sans pour autant préciser les caractéristiques de ce robot. En France, un peu moins d'une douzaine de fournisseurs de matériel pédagogique proposent une gamme de robots programmables aux formes, couleurs et modalités de programmation variées, offrant des potentialités pédagogiques et didactiques assez larges. Les fournisseurs qui ont bien voulu nous communiquer les pourcentages de ventes de robots nous permettent de dessiner des tendances nettes dans les choix effectués. Sans surprise, les robots distribués dans les écoles ne sont pas les mêmes que ceux que l'on peut trouver dans les lycées.

Cette étude auprès des fournisseurs de robots pédagogiques montre nettement que les BeeBot et les Bluebot sont extrêmement demandés. Cependant, ce choix est-il lié à des raisons didactiques portant sur les contenus mis en jeu à travailler en classe ? Les motifs qui orientent les choix d'achat des robots semblent être tout autres comme le suggère un des principaux fournisseurs :

Sur les seuls aspects commerciaux et *business*, mais qui orientent tangiblement le marché et les pratiques pédagogiques, il faut noter que les prescripteurs institutionnels (le réseau Canopé, les directions académiques au numérique éducatif, les inspections) font plus volontiers la promotion de robots de sources lointaines ou bien subventionnés par des institutions. La raison est de ne pas sembler favoriser un acteur privé local. Aussi les seuls robots vraiment plébiscités et poussés viennent de l'étranger. Il ne faut pas que l'on identifie qui s'enrichit sur la production de matériel pédagogique.



Ce discours semble affirmer que le choix des robots est influencé par des acteurs extérieurs aux écoles. Ce fournisseur nous donne les trois critères qui sont retenus par ces prescripteurs.

Le choix des robots et systèmes que les enseignants utilisent est guidé :

- bien entendu par les qualités intrinsèques des produits dans leur contexte d'utilisation (pertinence pédagogique, possibilités de scénarios conformes aux attentes, fiabilité entre les mains des élèves, facilité de prise en main par les profs et les élèves, qualité et quantité des ressources pédagogiques associées, etc.);
- par le « look » des produits (qui semble être déterminant) : les couleurs et le design d'un robot pour le primaire ne sont pas les mêmes que ceux d'un système pour l'université. Une évidence !
- par le prix : les robots chinois et occidentaux issus de programmes subventionnés ou soutenus par des institutions se taillent la part du lion en terme de diffusion.

La qualité de l'artefact, sa forme, sa couleur et son prix sont les principaux critères avancés par les fournisseurs, et retenus par les acheteurs, pour choisir un robot. Ce sont des critères principalement liés aux aspects de l'artefact plutôt qu'à son rôle dans l'enseignement de contenus.

Du côté des enseignants interrogés dans l'enquête, le choix des robots utilisés en classe semble être délégué à un tiers (figure 1). Seulement 7 % des enseignants déclarent choisir eux-mêmes les artefacts pour leur classe contre 69 % qui font confiance à l'équipe pédagogique de l'école ou à des conseillers ou référents technologies de l'information et de la communication pour l'éducation (TICE) extérieurs à leur école.

Les enseignants se voient ainsi imposer un artefact pour enseigner l'informatique dans leur classe, sans toujours connaître les critères qui ont guidé ce choix.

### **Modèle pour caractériser les choix**

Le choix d'un artefact en classe a une influence sur les contenus enseignés. Notre cadre théorique et les résultats présentés nous conduisent à proposer un modèle pour caractériser les profils des enseignants dans cette activité de choix d'artefact. Ce modèle présenté et discuté lors d'un colloque sur l'école primaire du XXI<sup>e</sup> siècle (Gruiger, 2021), conduit à examiner l'activité de choix

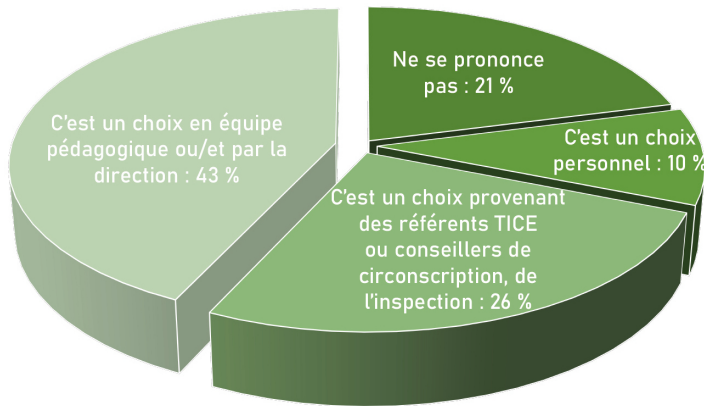


FIGURE 1 – Pratique déclarée de choix des artefacts.

d'artefact au regard des contenus pris en compte ou non. Cette activité est en effet à la fois une activité technique et didactique. D'une part, elle s'appuie sur les critères de l'artefact, d'autre part elle oriente les contenus qui seront abordés en classe. Enfin, elle peut être individuelle mais aussi collective. Le modèle est élaboré sur les trois composantes suivantes issues du cadre de la technicité (Combarnous, 1984) qui orientent le choix :

- une rationalité technique et didactique avec une réflexion propre à l'artefact technique incluant des visées d'apprentissage disciplinaires (informatique ou autres) ou non ;
- une composante matérielle liée aux constituants technologiques de l'artefact avec un choix en fonction des composants de l'artefact (capteurs, actionneurs. . .) d'une part, et de ses aspects, d'autre part (forme, couleur. . .) ;
- une composante d'influence sociale. Les prescripteurs, le cadre professionnel, les décideurs orientent les pratiques. Cette composante sociale interroge le rôle des différents acteurs dans le choix des artefacts. Une personne tierce, comme nous l'avons identifié, peut mener l'activité de choix avec une rationalité propre sans pour autant que l'enseignant en mesure les potentialités didactiques.

En fonction de sa genèse instrumentale personnelle et professionnelle (Haspekian, 2011), un enseignant développera une expertise soit du point de vue technologique seul, soit du point de vue didactique des apprentissages informatiques en jeu, tenant compte des aspects technologiques. Il sera également en mesure de reproduire des situations envisagées et pensées par d'autres avec des visées de construction de contenu ou d'animation scolaire centrées sur la réussite de l'activité.

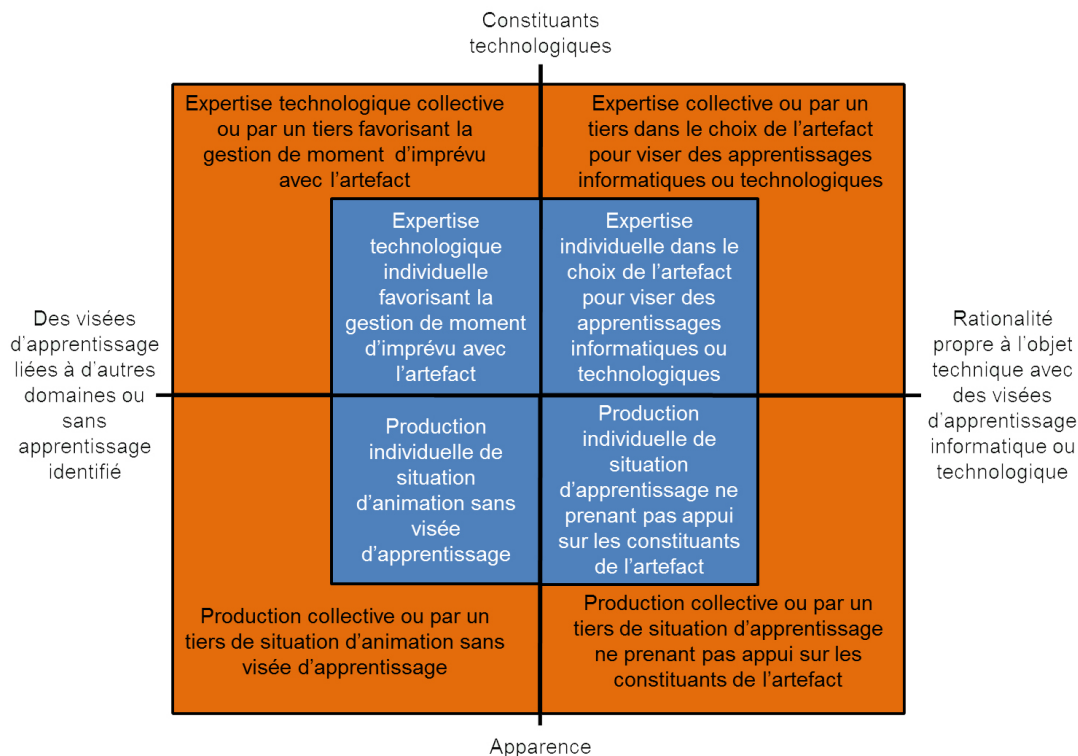


FIGURE 2 – Modèle de registres d'activité de choix d'un artefact avec une visée d'apprentissage.

Le modèle permet ainsi de situer une activité de choix d'un enseignant, pour un artefact dans un usage scolaire, dans huit registres possibles (figure 2) et d'interroger les contenus associés.

Ce modèle rend compte des trois dimensions de l'analyse, en écrasant sur une figure plane ce qui est en réalité un volume, en trois dimensions : un choix guidé par les visées d'apprentissages informatiques ou non (en abscisse) ; par des constituants technologiques ou l'apparence (en ordonnée) ; et guidés par des raisons personnelles ou collectives (en profondeur ou du centre vers l'extérieur).

Utilisons ce modèle pour analyser une activité de choix d'artefact et situer un type de pratique. Prenons l'exemple d'une enseignante de MS qui a déclaré, dans le questionnaire, que le robot le plus intéressant pour enseigner est le BeeBot puisqu'il est « très ludique pour mener les séances grâce à l'animal représenté ». L'apparence de l'objet semble orienter le choix de celui-ci. Elle ajoute que ce robot a été choisi collectivement par l'équipe pédagogique et qu'elle envisage de mettre en place une activité de « jeu de chasse aux trésors » avec ce robot. Le modèle de registre d'activité

de choix permet de situer cette pratique déclarée dans la zone inférieure gauche et en arrière-plan. Cette pratique est plutôt orientée vers des visées d'apprentissage éloignées des savoirs en informatique, tout comme celle de cet enseignant de cours élémentaire de 2<sup>e</sup> année (CE2) avec une expérience de 22 ans qui déclare utiliser les robots « Ozobot et Thymio ». Selon ce dernier, « les deux permettent des raisonnements complexes et la mise en place de situations problèmes », qui ne sont pas des savoirs spécifiquement liés à l'informatique. Cependant, il minimise ses propos en affirmant qu'il « ne connaît pas la pertinence des autres robots ». Car, le choix de matériel n'est pas personnel, c'est un choix de la collectivité territoriale.

Un seul enseignant de notre corpus disposant de robots dans sa classe déclare vouloir mettre en place du codage avec ces élèves MS et GS alors que cet enseignant n'a pas choisi le robot. Cependant, son parcours universitaire semble expliquer cette orientation. En effet, il s'agit de l'unique enseignant qui détient un master « technologique ». Sa pratique déclarée se situe alors dans la partie supérieure droite de notre modèle.

Ce modèle nous a permis de situer l'activité de choix des enseignants du corpus globalement du côté gauche, c'est-à-dire que les robots ne sont pas identifiés par les enseignants comme des outils pour viser des apprentissages en lien avec l'informatique.

## Conclusion

En conclusion, les analyses des entretiens comme des questionnaires montrent que les enseignants ont conscience de certains enjeux concernant un enseignement de l'informatique, notamment, pour certains, celui de la programmation. L'analyse des choix didactiques des enseignants confirme, si besoin était, que les enseignants ne sont pas des exécutants passifs des prescriptions et instructions officielles. Les enjeux didactiques sont inextricablement mêlés à des considérations institutionnelles, matérielles ou personnelles, qui expliquent la mise en œuvre ou non des programmes dans les classes. Dans ces rapports de force, la composante personnelle semble jouer plus fortement que les injonctions institutionnelles.

En effet, d'une part, ce qui s'apprend en programmant n'est pas toujours clair. Il peut s'agir de compétences non disciplinaires (la résolution de problèmes) ou disciplinaires mais en dehors de l'informatique, des notions spécifiquement informatiques étant très rarement évoquées spontanément. D'autre part, les enseignants rencontrés ont tendance à identifier des enjeux liés à des usages et des attitudes à développer chez leurs élèves plutôt qu'à des apprentissages notionnels. Ils s'attribuent alors un rôle minime

pensant que les élèves utilisent déjà des matériels informatiques dans le milieu familial et disposent, de ce fait, de plus d'habiletés manipulatoires qu'eux-mêmes.

Cette minimisation, à la fois de cet enseignement et du rôle qu'ils peuvent y jouer, se retrouve dans le fait que les enseignants délèguent souvent le choix d'achat de matériel à la collectivité territoriale, à l'équipe pédagogique ou encore à la circonscription d'inspection, se préoccupant peu de l'incidence sur les notions abordées avec les élèves.

Les résultats présentés dans ce chapitre conduisent également à proposer un modèle de registres de « l'activité de choix » d'artefact qui questionne les enjeux didactiques conditionnés par les choix d'artefacts.

Ce chapitre, après celui présentant les prescriptions, recommandations ou discours institutionnels, permet de contextualiser les pratiques enseignantes et les apprentissages des élèves, présentés dans les chapitres suivants.

### Recommandations

Ces résultats permettent d'évoquer plusieurs pistes pour améliorer les pratiques envisagées d'enseignement de l'informatique. Premièrement, nos résultats confirment la nécessité d'une formation des enseignants aux enjeux de savoir liés à l'informatique, afin que la mise en œuvre effective d'enseignements informatiques ne repose pas principalement sur le sentiment de compétence personnelle des enseignants et l'autoformation. Nous recommandons que ces formations ne soient pas construites sur un modèle *top-down* transmissif mais partent des pratiques effectives actuelles. Par exemple, nos résultats identifient le fait d'être actif face à la machine (par opposition à une position de récepteur passif) comme étant une attitude recherchée chez beaucoup d'enseignants, sans qu'eux-mêmes ne l'identifient nécessairement. Une telle position pourrait être travaillée en lien avec des contenus en informatique et plus particulièrement pour construire une pensée informatique, ce qui peut être défini comme un objectif de formation.

En second lieu, un accompagnement des enseignants dans le choix du matériel didactique pour enseigner l'informatique permettrait également de recentrer sur des enjeux de savoirs les choix d'équipement des classes et des établissements.

Troisièmement, les enseignants ressentent souvent les effets délétères de discours institutionnels contradictoires (ou perçus comme tels). Il y a par exemple des discours contradictoires entre la nécessité de former les jeunes à l'informatique et celle de protéger les enfants de l'exposition aux écrans, ou encore entre cette nécessité et les discours présentant les jeunes générations comme des *digital natives*. Une première recommandation serait alors de clarifier les moyens possibles, en incluant, notamment, le recours à des dispositifs sans écran (comme la robotique pédagogique et l'informatique débranchée). Une seconde recommandation, serait que ces discours soient rendus plus cohérents par les institutions scolaires. Pour aider les enseignants à mieux appréhender ces nouveaux contenus, il serait nécessaire par exemple que soient mieux explicitées les visées des enseignements informatiques pour les élèves, au regard des discours médiatiques sur les soi-disant « natifs numériques ».

### Références

Combar nous, M. (1984). *Comprendre les techniques et la technicité*, Éditions sociales.

Couderette, M., Haspekian, M., et Kiwan, M. (2023). « Algorithmique, programmation, informatique », dans *Les prescriptions en mathématiques : un premier repérage* [présentation orale], Séminaire IE CARE, Paris, France.

Daunay, B., et Reuter, Y. (2013). « Penser et problématiser les contenus disciplinaires : un enjeu fondamental pour les didactiques », dans B. Daunay, Y. Reuter et A. Thépaut (dir.), *Les contenus disciplinaires. Approches comparatistes*, p. 21-34, Presses universitaires du Septentrion. <https://doi.org/10.4000/books.septentrion.15740>

Delcambre, I. (2013). « Contenus d'enseignement et d'apprentissage », dans Y. Reuter (dir.), *Dictionnaire des concepts fondamentaux*

des didactiques, 3<sup>e</sup> éd., De Boeck, p. 43-48. <https://doi.org/10.3917/dbu.reute.2013.01.0043>

Fluckiger, C. (2017). À la convergence entre info-doc et informatique/numérique : quels contenus, quel enseignement? *Journée départementale de l'Association des professeurs documentalistes de l'académie de Nantes du 4 janvier 2017*.

Grugier, O. (2020). Éducation technologique dans des classes de maternelle. Apprentissages premiers dans l'utilisation et la compréhension d'un artefact robotisé, *Recherche en didactique des sciences et des technologies*, vol. 22, p. 61-92. <https://doi.org/10.4000/rdst.3328>

Grugier, O., Fluckiger, C., et Haspekian, M. (2021). Enseignement de l'informatique à l'école : choix des artefacts par les enseignants, *Colloque L'école Primaire au 21<sup>e</sup> siècle*, Cergy, France. <https://hal.science/hal-03378486v1>

Haspekian, M. (2011). « The co-construction of a mathematical and a didactical instrument », dans M. Pytak, E. Swoboda, et T. Rowland (dir.), *Proceedings of the Seventh congress of the European Society for Research in Mathematics Education*, p. 2298-2307. <https://hal.science/hal-01273866v1>

Reuter, Y. (2011). « Penser la perspective didactique : la question de l'articulation entre disciplinaire, pédagogique et scolaire », dans B. Daunay, Y. Reuter et B. Schneuwly (dir.), *Concepts et méthodes en didactique du français*, Presses universitaires de Namur, p. 15-40.

Reuter, Y. (2013). *Un déplacement théorique en didactique(s) : le vécu disciplinaire des élèves, Souffrances, plaisirs et autres émotions des enfants à l'école*, Journée d'étude organisée par le laboratoire EMA, Cergy-Pontoise, France.

Robert, A., et Rogalski, J. (2002). Le système complexe et cohérent des pratiques des enseignants de mathématiques : une double approche, *La revue canadienne de l'enseignement des sciences, des mathématiques et des technologies*, vol. 2, n° 4. <https://doi.org/10.1080/14926150209556538>

Pour citer ce chapitre :

Fluckiger, Cédric, Grugier, Olivier, et Haspekian, Mariam (2024). « Pratiques envisagées des enseignants pour un enseignement de l'informatique à l'école primaire », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 51-68. <https://doi.org/10.53480/2024iecare03q/>

# Quelles ressources pour enseigner l'informatique dans le premier degré?

## Une étude de cas

# 4

Pascale KUMMER-HANNOUN<sup>1,2</sup>  
Georges-Louis BARON<sup>2</sup>

1. Sorbonne Université, Faculté des Sciences, UFR d'Ingénierie, 75005 Paris, France

2. Université Paris Cité, Laboratoire EDA, 75006 Paris, France

L'informatique a trouvé un certain droit de cité à l'école primaire en 2015, à l'issue d'un long parcours marqué de mouvements de flux et de reflux de l'intérêt éducatif pour la programmation (Baron et Drot Delange, 2016). Au sein du projet « Informatique à l'école : conceptualisations, accompagnement, ressources » (IE CARE), nous avons décidé de nous centrer sur les pratiques ordinaires des enseignants intégrant la robotique à l'école primaire en étudiant comment ils choisissent et adaptent les ressources sur lesquelles ils s'appuient. Ces questions nous ont paru d'autant plus importantes que les ressources permettant d'étayer un enseignement de robotique sont encore réduites. Leur pénurie est un des facteurs limitant la mise en œuvre d'activités de programmation, comme cela a été montré encore récemment pour le Canada (Stokes *et al.*, 2023).

Notre choix s'est porté sur les robots les plus simples (les BeeBot) qui ont l'avantage d'avoir seulement cinq commandes : avancer ou reculer (tout droit, pendant une seconde), pivoter de 90° à gauche et à droite, faire une pause d'une seconde. Jusqu'à 40 pas de programme peuvent être mémorisés. Une touche spéciale (GO) permet de lancer le programme et une autre (X) d'effacer la mémoire du robot. Ils ont fait l'objet de nombreuses recherches, dès l'école maternelle.

Komis et Misirli (2012) ont montré, après une recherche menée en Grèce avec des jeunes de quatre à six ans, que ces jouets programmables peuvent conduire au développement de « compétences relatives à des notions mathématiques, à la pensée algorithmique et aux stratégies de résolution des problèmes ».

Ils remarquent que, guidés par un scénario pédagogique, les élèves peuvent « construire des programmes séquentiels » (en utilisant des ressources comme des feuilles de programmation où sont



inscrites la suite des commandes constituant le programme) et les entrer sur le BeeBot dont la course renseigne ensuite sur la justesse du programme. Les commandes de base (AVANCE, RECULE, GO) sont facilement appropriées, mais la latéralisation est plus difficile. Ils concluent que « le développement des compétences de programmation (pensée algorithmique, séquence, notion de mémoire) nécessite une contextualisation adéquate au moyen de scénarisation pédagogique pour motiver de manière efficiente les jeunes enfants ».

Notre recherche, centrée sur une étude de cas avec des élèves légèrement plus âgés de six à huit ans (cycles 1 et 2-3), pose la question des ressources mobilisées par une enseignante qui introduit la robotique dans sa classe pour la première fois, et construit elle-même ses séances. Nous la suivons pendant deux années.

Nous nous sommes particulièrement intéressés aux questions suivantes :

- Quelles ressources l'enseignante choisit-elle pour soutenir son action ?
- Comment les utilise-t-elle pour enseigner la programmation ?
- Quelles sont les transformations qui surviennent d'une séance à l'autre ? Quelles sont les évolutions d'une année sur l'autre ?

## Méthodologie

L'enseignante observée a une solide expérience (plus de 20 ans) d'enseignement au primaire. Elle a fait des études scientifiques (licence de biologie), a un intérêt marqué pour l'innovation pédagogique et ne craint pas les prises de risques afférentes. Nous avons déjà travaillé avec elle dans le cadre d'un projet de sciences à l'école. Si elle a déjà souvent fait travailler ses élèves avec des tablettes, elle n'avait pas encore organisé d'activités en informatique proprement dite.

Comme nous souhaitons mener des observations en profondeur et acquérir une bonne compréhension de ce qui se jouait sur la durée, cette recherche exploratoire est centrée sur un seul cas. Il s'agit de documenter l'activité ordinaire dans une situation nouvelle pour l'enseignante, sans partir d'une idée *a priori* de ce qu'elle pourrait ou devrait être. Les outils d'observation et de récolte de données sont axés sur l'activité de l'enseignante avec les ressources et avec les élèves, et non sur l'activité de ces derniers.

## Contexte de l'observation en classe

En France, les programmes de 2015 pour les six à huit ans suggèrent l'introduction de notions de codage et décodage de déplacements dans le cadre des mathématiques et en découverte du monde (cf. le chapitre 2 « Quel enseignement de l'informatique dans la scolarité obligatoire en France ? Analyse des programmes et manuels »). Les manuels de mathématiques de ces niveaux ne consacrent qu'une page ou deux à l'étude des déplacements et codages des robots. Les enseignants sont amenés à recourir aux sites Internet qu'ils soient institutionnels, associatifs ou de collègues, pour y trouver d'autres idées.

Outre le manque de « ressources scolaires traditionnelles » consacrées à la programmation à ces niveaux scolaires, des besoins spécifiques liés aux contraintes de l'activité de robotique sont rarement satisfaits. Ces objets techniques ont un coût important pour le budget des écoles et un minimum d'espace est requis pour leur déplacement. Ici, les conditions étaient très favorables.

- La salle de classe est reliée à une pièce servant de bibliothèque, avec un revêtement de sol homogène, des livres, une longue table et des chaises où les élèves peuvent préparer leurs programmes.
- Un ensemble de six robots BeeBot a été prêté à l'école par le laboratoire EDA.

La première année (2019–2020), les robots sont utilisés en cours élémentaire 1<sup>re</sup> et 2<sup>e</sup> année (CE1 et CE2) par 22 élèves. La crise sanitaire n'a pas permis de poursuivre l'observation l'année suivante, puis en 2021–2022, l'enseignante a une classe de cours préparatoire (18 élèves). Elle conçoit elle-même sa progression, différemment suivant les niveaux.

Dans les deux cas, elle divise sa classe en deux demi-groupes, et forme des petits groupes de deux à trois élèves. Pendant qu'une moitié de la classe est en robotique dans la bibliothèque, l'autre est dans la salle principale (en autonomie pour les CE, sous la surveillance d'un assistant pédagogique en cours préparatoire (CP)). Nous avons pu observer sept des neuf séances en CE, réparties entre décembre et mars. Et en CP, 14 des 16 séances réparties entre novembre et juin. Les séances (de 40 à 50 minutes) ont lieu une fois par semaine en alternance, avec des interruptions pour raisons techniques (deux fois, les groupes de CP ont eu le temps de se succéder dans la même séance).

La structure des scénarios pédagogiques avec les BeeBot présente pour chaque niveau scolaire une phase de découverte du robot (deux séances par demi-groupe), puis une phase de défis (deux séances en CE, trois en CP), avec quelques différences :

- Une séance en classe entière est organisée à mi-parcours en CE à des fins d'institutionnalisation, mais non en CP.
- L'apprentissage des notions liées aux déplacements sur un quadrillage a été travaillé précédemment en mathématiques en CE (nous ne l'avons pas observé). En CP, il est inclus aux séances de robotiques (quatre séances par demi-groupes).

Dans ce chapitre, nous limitons l'analyse des ressources pédagogiques à celles utilisées pendant la phase de découverte des robots ainsi que pendant les activités de codage de déplacements qui précèdent. Celles utilisées pour les défis sont évoquées à la marge.

### **Choix effectués**

Il s'agit des choix méthodologiques faits pour l'observation des séances de classe et pour analyser les données recueillies.

### **Modalités d'observation**

Nous (nous deux, ou l'un de nous) avons observé à chaque séance différents groupes d'élèves, rempli une série de cahiers avec la description des principaux événements, recueilli du matériel pédagogique et des travaux d'élèves, et enregistré les interactions. Il s'agit de garder la trace des moments où l'enseignante introduit une ressource et la présente, des consignes qu'elle ajoute et de ses modes de guidage (les élèves, eux, ne sont pas toujours audibles). Des entretiens semi-directifs avec l'enseignante en amont des observations et à la fin des séances ont été menés pour identifier par exemple d'où provenaient certaines ressources, ainsi que les difficultés rencontrées. Il est arrivé que lors d'échanges informels en fin de séance nous lui fassions une suggestion en réponse à une demande.

### **Méthode pour l'analyse des données**

Pour l'analyse, nous avons opéré une distinction entre différents types de ressources utilisées :

- celles qui sont déjà en place et sur lesquelles l'enseignante compte;
- celles qu'elle produit avant les séances (par sélection et transformation d'autres ressources), pour organiser l'activité des élèves;
- et enfin celles qui apparaissent en cours de séance.

Toutes les ressources mobilisées sont susceptibles d'être adaptées ou modifiées d'une séance sur l'autre, et d'une année à l'autre. Ce choix de description permet aussi de suivre la chronologie des activités proposées.

## Utilisation d'un large spectre de ressources

L'enseignante a eu recours à une grande diversité de ressources, que nous allons considérer en respectant la catégorisation présentée ci-dessus.

### Ressources disponibles

Parmi les ressources « déjà là », il y a celles qu'on pourrait qualifier de ressources d'infrastructure et qui ont un rôle essentiel : la disposition de la salle (arrangée par l'enseignante), la présence d'un assistant scolaire, les robots (qu'il aura fallu recharger avant chaque séance. . .), et les prescriptions officielles.

D'autres font partie du matériel habituel de la classe et n'ont rien de spécifique, comme le manuel de mathématiques, les outils utilisés habituellement par les élèves (papier / crayon / gomme / règle graduée) ainsi que différents objets présents qui ne semblent pas avoir de rapport direct avec l'activité de robotique mais seront parfois utilisés. Dans cette catégorie entrent les *kaplas* utilisés pour matérialiser le trajet du robot.

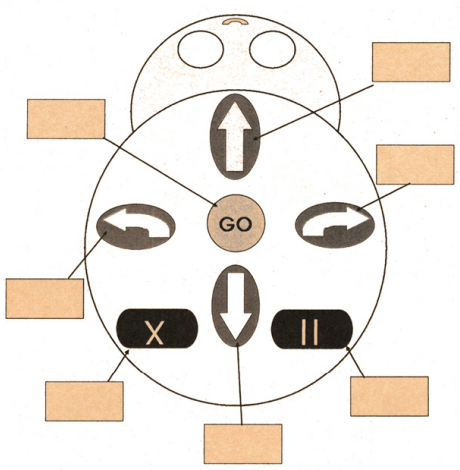
### Ressources créées spécifiquement pour la robotique

La phase de découverte des robots a eu pour principaux objectifs d'apprentissage de définir la notion de robot (« une machine qui exécute des ordres ») et le fonctionnement du BeeBot : la signification des commandes, prioritairement le rôle de la touche « effacer la mémoire » ainsi que la distinction entre tourner et pivoter, de façon à pouvoir coder un trajet avec virages.

Dans cette section nous présentons la « fiche initiale » préparée spécialement par l'enseignante la première année (en CE) et son lien avec ses pratiques habituelles. C'est une feuille A4 au format paysage, avec des inscriptions en noir sur un papier beige. Elle ressemble par son aspect aux autres supports fournis pour d'autres activités scolaires. Conforme aux pratiques habituelles de l'enseignante et donc aux habitudes des élèves, elle indique que la manipulation du robot, distribué en même temps que la fiche, n'est pas de l'ordre d'un jeu mais relève d'une activité scolaire, comprenant des tâches de lecture et d'écriture (figure 1).

**Étude d'un robot de sol : Bee-bot / Blue-bot**

**Robot (n, m) :** Un robot est une machine automatisée ou contrôlée à distance qui réalise des tâches dangereuses ou pénibles pour un humain. La robotique est l'étude et la réalisation de robots.  
[Vikidia, l'encyclopédie des 8-13 ans]



**Étape 1 :** Allume et manipule le robot de façon à pouvoir identifier, avec tes camarades, les différentes fonctions du robot, indiquées par les touches sur son dos. **Remplis** le schéma ci-contre.

**Étape 2 :** Détermine quelle est la distance d'un pas du robot ?  
un pas du robot = ..... cm

**Étape 3 :** Maintenant que tu maîtrises son fonctionnement, fais-lui réaliser les figures ci-dessous. **Écris le programme** de son déplacement (flèches).  
**Place** un point rouge sur chaque figure pour indiquer le début du programme de déplacement.

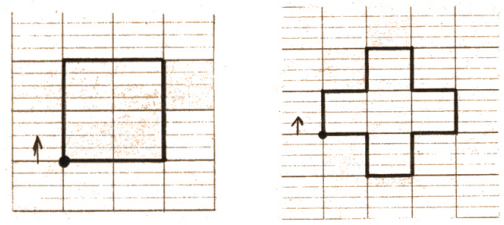


FIGURE 1 – La fiche initiale, consignes et schéma à compléter en CE1-CE2.

Elle comprend deux parties ; à gauche, un schéma du robot sur lequel on reconnaît les commandes ; à droite, trois consignes :

- découvrir la signification des touches du robot, en le faisant fonctionner ;
- mesurer la longueur du pas du robot ;
- programmer le robot pour qu'il se déplace en reproduisant les deux figures représentées, un carré puis une croix.

Cette fiche a plusieurs rôles : support de consignes et d'écriture pour les élèves, acquisition de traces pour les évaluer.

La confrontation de cette fiche au discours de l'enseignante pendant les interactions de classe ainsi qu'au manuel de mathématiques et aux textes officiels révèle que les trois consignes sont choisies en relation avec des activités de mathématique réalisées peu de temps avant, ou prescrites à ce niveau scolaire.

Les commandes de déplacement du robot ressemblent aux flèches utilisées par les élèves la semaine précédente lors d'un exercice de codage sur quadrillage du fichier de maths. D'ailleurs, certains élèves remplissent les cases associées aux commandes représentées

sur la fiche, avec par exemple « à droite », « à gauche » sans même allumer le robot pour tester ses déplacements.

La mesure de la longueur du pas est reliée à la « mesure de segment », compétence présente dans les programmes, et a déjà fait l'objet d'exercices. L'enseignante le rappelle pour raviver les souvenirs des élèves sur la façon de procéder. Utiliser une règle, caler le zéro de la règle au début du segment, lire sur la règle la graduation qui se trouve en face de la fin du segment, etc.

Et enfin, les parcours de la dernière consigne sont basés sur des formes géométriques. Le carré est, d'après les textes officiels, une forme à savoir reconnaître et reproduire. Pour la croix, une explication peut être que l'enseignante a cherché une forme plus difficile et compatible avec les seules rotations à angle droit permises par le robot (*exit* le cercle, le triangle. . .).

Les séances de robotique s'inscrivent donc dans une continuité pédagogique de temps, d'espace et de méthode, en relation forte avec le programme de mathématiques. La fiche s'est révélée très ambitieuse et a demandé deux séances à la plupart des groupes. La mesure du pas, qui nécessite de repérer la position d'un même point du robot avant et après son déplacement, a posé beaucoup de problèmes. Le parcours en forme de croix (plus de 20 pas et rotations dans les deux sens) s'est révélé bien plus difficile que le carré.

L'écart entre le temps nécessaire, projeté par l'enseignante, et le temps effectif l'a poussée à ajuster son aide en cours de séance et à s'interroger sur les obstacles rencontrés.

### **Ressources imaginées en cours de séance pour pallier quelques obstacles**

Ces ressources sont intéressantes car elles correspondent à celles qui manquent, ou du moins à un besoin non prévu. Ici, elles sont de l'ordre de l'analogie avec d'autres systèmes techniques, ou de gestes.

#### **Pour les touches « X » et « II » et le « pivot », des analogies**

Lors de cette première séance, l'identification de la fonction des touches « X » (effacer la mémoire) et « II », (faire une pause d'une seconde) n'est pas immédiate. Les tests que les élèves peuvent faire (appuyer sur la touche « X », ou sur « X » puis sur « II », ou l'inverse, avant d'appuyer sur « GO ») n'aboutissent pas à la solution, car ils produisent le même résultat : le robot n'avance pas. Et si les élèves n'ont pas effacé la mémoire avant d'entrer un nouveau programme, le robot semble faire « n'importe quoi ».

Pour débloquer les groupes, l'enseignante apporte des aides de différentes natures. Elle rappelle qu'un robot ne peut « qu'exécuter les ordres qu'on lui a donnés ». Ensuite elle leur demande s'ils ont déjà rencontré ces symboles sur d'autres appareils, téléphones, vidéos. . . Les élèves trouvent la signification de la touche « pause » du robot grâce à cette analogie pourtant trompeuse, car c'est une instruction exécutable et non une commande externe. Elle sera utilisée pour les défis.

Si la touche « X » reste mystérieuse pour un binôme, elle transforme l'activité libre en activité cadrée, avec la consigne : « demande au robot d'avancer de deux pas » et rappelle l'existence de la « mémoire » du robot. Alors, les élèves peuvent vérifier si les déplacements du robot sont ceux attendus. S'ils restent dérouterés par le résultat, l'enseignante suggère d'appuyer sur « X » et de recommencer. Ceci suffit aux élèves pour comprendre que la mémoire du robot a été effacée. Mais il reste une ambiguïté : certains élèves, pour effacer un programme appuient sur cette touche plusieurs fois, comme avec la commande « suppr » d'un clavier d'ordinateur ou de smartphone qui efface les caractères un par un, alors que la mémoire BeeBot s'efface en une fois.

L'enseignante doit aussi insister sur la différence entre « tourner en avançant comme le font les voitures », et le mouvement de « pivot sur place » du robot. Cette fois, c'est une analogie par la négative.

### **Coder en référentiels relatifs, introduction de gestes outils**

La première figure à faire exécuter au robot est un carré. Pour l'obtenir, il doit avancer puis pivoter, quatre fois de suite. Les commandes de déplacements sont interprétées par le robot en fonction de sa propre orientation : ainsi « avance d'un pas » fait avancer le robot droit devant lui quelle que soit sa position dans la pièce. C'est ce qu'on appelle une référence relative. Pour l'élève qui code, cela suppose qu'il se décentre. En effet si le robot avance vers lui, la touche « pivote à droite » fait tourner le robot vers la gauche de l'élève. Mais les mouvements vers l'avant posent aussi problème, car « avance d'un pas » est codé par une flèche vers « le haut » de la feuille qui signifierait en repère absolu « va vers le nord » (ou « le haut »).

Pour aider les élèves à se repérer, l'enseignante place son doigt au point de départ, et demande aux élèves dans quelle direction doit aller le robot. Si la réponse est correcte, l'élève trace la flèche correspondante. Après validation, l'enseignante déplace son doigt à la position suivante. Si l'élève est perdu, elle déplace aussi le robot à la main, le long d'un carré imaginaire sur le sol.

Dans un des groupes, une élève a l'idée de se déplacer en tenant le robot dans le même sens qu'elle, ainsi elle ne se trompe plus pour indiquer les commandes à son binôme qui écrit le code.

Ces gestes qui seront repris par d'autres groupes, constituent des micro-ressources.

Ce sont plusieurs exemples d'agilité de l'enseignante qui tire profit des objets présents dans la salle et des objets du quotidien, présents dans l'esprit des élèves, et ajuste son guidage, en passant de consignes très ouvertes à des tâches plus simples ou plus contraintes.

Ayant constaté que plus de la moitié des élèves ont des codes erronés ou absent pour le carré, et qu'un seul élève a su programmer les deux trajets (incomplet pour la croix), elle introduit alors une séance de correction collective avant de passer aux défis.

Nous reviendrons plus loin sur les difficultés résiduelles à l'issue de ces séances.

## Quelles transformations des ressources ?

La première année, des ajouts de nouvelles ressources sont fréquents pour résoudre certains problèmes. Par exemple, devant la difficulté des élèves à visualiser le déplacement des robots directement sur le sol, elle ajoute pour certains défis des plaques de bristol quadrillées aux dimensions du pas du robot ou propose des trajets matérialisés par des *kaplas*. Elle fabrique aussi des bandes de programmation avec des cases pour faciliter l'écriture des codes (cet ajout vient d'une proposition que nous lui faisons<sup>1</sup>).

1. L'usage de bandes algorithmiques a été préalablement observé par Spach (2017).

## Évolution des ressources en CP par remplacement et simplification

Avec les élèves de CP, elle change de méthode et crée un nouveau système de ressources, basé sur plusieurs remplacements. Pour la phase de découverte des fonctions du robot, elle supprime la fiche initiale, mais s'en inspire, nous verrons comment à la section suivante. Pour construire les prérequis de codage de déplacements, elle n'utilise pas le fichier de maths (bien qu'il y ait aussi des quadrillages dans le fichier de maths de CP), et se base sur le « jeu de l'enfant robot » popularisé par Greff (1998) et ensuite présenté en ligne par Dufлот<sup>2</sup>.

2. <https://www.youtube.com/watch?v=9AtmJ9mTaB0>



### Construction d'un tapis de sol et ajouts d'étapes

Le jeu de l'enfant robot est mis en œuvre sur un grand tapis de sol créé par l'enseignante, il contient une grille dont les cases ont la dimension des pieds des élèves. Quelques cellules mobiles, bleues ou vertes, représentent une rivière ou une forêt que les élèves doivent éviter. Des cellules marron représentent des ponts pour passer par-dessus la rivière. Un repère est placé pour la case départ, un plot sur la case d'arrivée. Un élève joue le rôle du robot, il doit suivre les instructions données par son binôme.

La progression s'effectue en étapes (sur plusieurs séances) avec le tapis de sol :

- une bande comportant un programme simple est fournie et il est demandé aux élèves de procéder à un décodage. Celui-ci s'effectue en deux temps (lecture du programme puis traduction en action);
- des erreurs sont introduites dans les bandes de codes pour amener les élèves à déboguer le programme;
- les élèves doivent imaginer un trajet, coder des bandes vierges (et effaçables), tester le programme avec leur binôme, puis ces bandes codées sont testées par d'autres élèves.

L'enseignante se rend compte qu'elle doit matérialiser les cases sur la bande et préciser le sens d'écriture; on est en décembre et les élèves (qui sont alors au début du CP) ont du mal à écrire les flèches et à se relire. Elle finit par construire un mini robot en lego et reproduit le paysage quadrillé sur une feuille A4, pour aider les binômes à imaginer le parcours et le coder. Mais ils auront du mal à s'organiser et à se répartir les tâches, d'autant qu'elle ne leur dit pas de représenter leur trajet sur la feuille.

L'ensemble de ces ressources représente un important travail de fabrication en amont, soutenu par le modèle proposé par Duflot dont il s'inspire fortement. L'enseignante ajoute des ressources outils qui n'existaient pas la première année, et continue à les perfectionner d'une séance à l'autre, mais introduit aussi de nouvelles activités comme le débogage.

### Remplacement de la fiche initiale

La fiche initialement utilisée ayant montré ses limites, l'enseignante ne s'en sert pas en CP. À la place, un guidage oral très directif, menant à des micro-tâches aboutit rapidement à une définition des commandes du robot, en particulier pour l'instruction « effacer la mémoire » nommée aussi par les élèves « oublie tout ». Les deux trajets de la fiche sont remplacés par quatre parcours plus simples avec repères au sol :

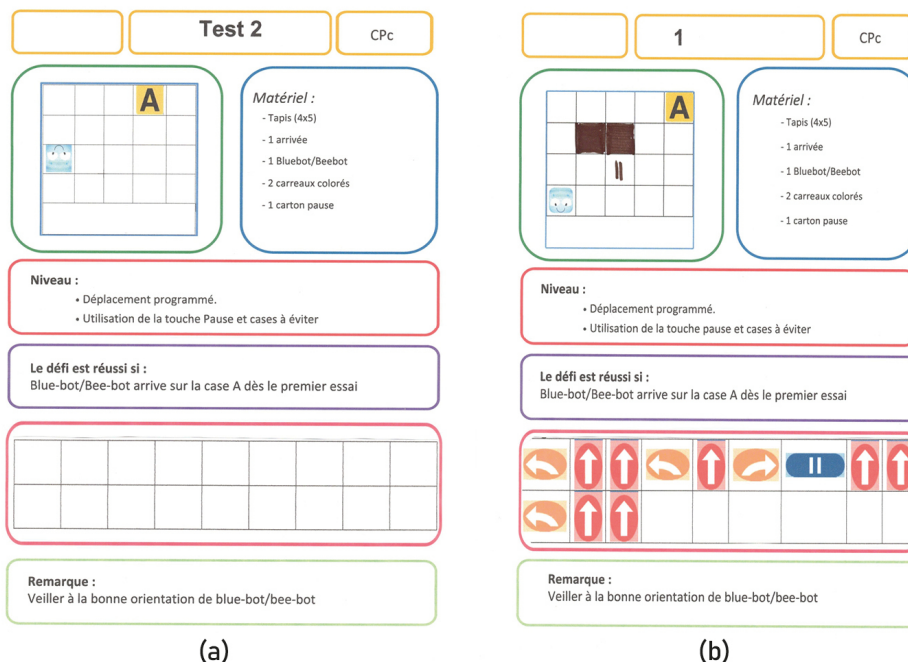


FIGURE 2 – Matériel fourni pour les défis, avec bandes de programmation : (a) à remplir, (b) déjà codée par l'enseignante.

- trois trajets en kapla : un couloir droit, un en L, et un carré (autour duquel le robot doit tourner) ;
- et un trajet libre (à définir entre deux cases imposées) sur une grille  $4 \times 5$ .

L'enseignante a supprimé l'étape délicate de mesure du pas et indique qu'il est légèrement plus grand que la longueur d'un kapla. Elle veut aussi leur montrer l'avantage de la grille, ajustée à la longueur des pas, pour visualiser le trajet du robot. Ces activités, en particulier la grille et le couloir tout droit, sont assez vite réussies par la plupart des binômes, qui programment directement sur le robot.

Elle proposera à nouveau l'usage de la bande de codage à remplir pour programmer les trajets des défis, (ce dont les élèves ne se sont en général pas emparés), et finira par l'imposer devant les difficultés persistantes et la nécessité pour elle de détecter la nature des erreurs. Elle ajoutera même en dernière séance une phase de décodage en fournissant une séquence codée utilisant des symboles identiques à ceux du robot. De cette façon, elle divise le problème à résoudre en plusieurs sous-problèmes.

## Identification et traitement de problèmes didactiques

Certaines difficultés sont liées à la technologie elle-même ou aux exercices de déplacement choisis dans les ressources disponibles. D'autres viennent de l'écart entre le fonctionnement du robot et les concepts mathématiques travaillés.

### Difficultés liées à l'objet technique

Le robot n'avance pas d'une longueur donnée mais pendant une certaine durée, ses moteurs étant synchronisés. Lorsque, pour une raison quelconque, les roues glissent, ou frottent, il ne se comporte pas comme il l'aurait dû (par exemple, la mesure de la longueur du pas donne des résultats différents suivant les supports). En outre, la métaphore de la rotation est trompeuse : le robot pivote sur place de 90 degrés (le moteur d'un côté s'arrête, l'autre continue de tourner). Ce comportement aussi dépend de la surface. L'enseignante a par la suite essayé d'éviter ce problème en s'assurant que les robots se déplaçaient sur le même type de surface. Cependant, les problèmes ont persisté et il est devenu courant, lorsque le programme était impeccable mais que le robot « buggait », de le pousser doucement pour qu'il puisse atteindre le but.

### Difficultés liées aux écarts entre les notions mathématiques prescrites et les déplacements du robot

Compte tenu des problèmes décrits ci-dessus, la commande « avance d'un pas » supposée compatible avec les opérations sur des nombres entiers, prescrites au cycle 2, ne l'est pas, car il est presque impossible d'avoir un sol parfaitement régulier qui donnerait le pas constant d'un « robot idéal ».

### Difficultés liées à l'appui sur les manuels de mathématiques

La première année, en introduction à la robotique, l'enseignante s'est appuyée sur les ressources existantes, tout particulièrement en mathématiques. Ces manuels proposent, en suivant les prescriptions, d'étudier les déplacements dans deux référentiels (référentiel absolu et référentiel relatif) et proposent des exercices qui adoptent l'une ou l'autre perspective.

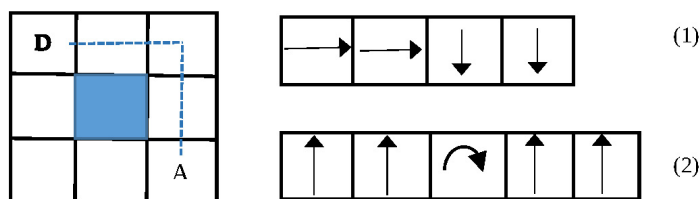


FIGURE 3 – Codage en référentiel absolu (1), centré sur le robot ou relatif (2).

Par exemple, la figure 3 montre deux façons possibles de coder une trajectoire très simple de D (départ) à A (arrivée).

Le premier chemin utilise un référentiel absolu, en utilisant des directions fixes (se déplacer deux fois vers l'est puis deux fois vers le sud) les directions des flèches sont semblables à celles du trajet dessiné entre D et A. Le second référentiel est relatif et reflète les commandes qui doivent être assignées au robot. Ce codage est plus difficile que le premier. Il exige que l'élève se décentre. De plus la rotation est une commande qui n'existe pas en référentiel absolu. Ces deux systèmes symboliques sont incompatibles, bien que très similaires.

### Nature des confusions en CE

L'analyse des codages réalisés par les élèves de CE pour le trajet carré montre que certains ont codé en référentiel absolu, ou ont mélangé les deux systèmes de codage. Par ailleurs, leurs flèches de pivotement ont un tracé rectiligne et non pas courbe.

Or, il y a eu en amont de la robotique des activités basées sur le manuel de mathématiques de la classe de CE1 (*À portée de maths*, 2016). Ce manuel proposait une seule page d'exercices en référentiel relatif (sans donner d'indication sur le référentiel utilisé). Il s'agissait de coder un robot-tortue, pour lequel on peut définir un angle de rotation, modalité très différente du déplacement des BeeBot. C'est pourquoi l'enseignante n'a pas choisi ces exercices mais a privilégié des exercices de codage et décodage de déplacements en repère absolu, avec des flèches, sur un quadrillage. L'apprentissage du codage en référentiel absolu avant de programmer des robots a pu favoriser la confusion dans l'esprit des élèves.

### Nature des confusions en CP

Le manuel de CP, plus récent (*Pour comprendre les maths*, 2018) mentionne le choix du référentiel (écrit en très petits caractères, dans un petit encadré). Il présente deux types de déplacements travaillés pour représenter ou coder un chemin dans une grille.

Le premier utilise un référentiel absolu avec des codes couleurs (un pas vers le mur rouge se code par  $n$  points rouges) tandis que l'autre, en référentiel relatif, est semblable au déplacement des BeeBot avec des flèches similaires aux touches des robots (sans marche arrière).

L'enseignante n'utilise pas ces exercices car elle trouve les codes couleurs trop éloignés des flèches, et comme l'exercice avec le robot est situé en fin de manuel, elle estime que cela indique qu'il n'est pas utilisable en début d'année. L'activité basée sur « le jeu du robot » a conduit à des confusions similaires. Dans la vidéo réalisée par Duflot, les deux possibilités de référentiels sont proposées. En référentiel relatif, le changement de direction s'effectue par quarts de tour du corps (vers la gauche ou la droite), tandis qu'en référentiel absolu ce sont des déplacements latéraux (un pas vers la gauche ou la droite sans pivotement), le corps est donc toujours orienté dans le même sens. Or, c'est ce type de déplacement que l'enseignante produit en faisant une première démonstration du jeu du robot, qu'elle corrige dès la deuxième séance. La confusion ne reste présente que chez quelques élèves. Et contrairement aux élèves de CE, ils ont tous su tracer des flèches arrondies pour les pivots.

## Discussion

L'analyse des ressources produites par l'enseignante observée montre comment elle a réussi à « scolariser » l'informatique en l'intégrant aux autres matières qu'elle enseigne, ainsi qu'à ses habitudes et pratiques pédagogiques. L'utilisation du fichier de mathématiques et la création de la fiche initiale en sont des exemples forts.

Les transformations apportées aux premières ressources ont mis en lumière la façon dont l'enseignante, en s'appuyant sur les difficultés des élèves, fait évoluer sa représentation de leurs besoins. Il en résulte des ressources qui s'écartent progressivement de l'ancrage mathématique, et dont la fonction évolue. Nous avons identifié différents mouvements :

- l'abandon de la fiche initiale, qui structure la séance en étapes mais ne contient pas d'outil pour dépasser les obstacles rencontrés, est un moment clé. Nous faisons l'hypothèse que le temps passé à les identifier et chercher à débloquer les élèves est un moteur de transformations ;
- la création de ressources intermédiaires, visant à prendre en charge les difficultés repérées (bandes de programmation, grilles de sol), apparaît dès la première année. Leur fonction est tout autre, nous sommes passés de ressources « consigne » à des ressources « instrument » (Nogry et Spach, 2022) ;

- et le découpage des tâches en sous-tâches, l'introduction de décodage, incluant du débogage sont des phénomènes qui apparaissent durant l'année de CP, corrélés à l'âge des élèves et favorisés par les possibilités offertes par le « jeu du robot ». Ils occasionnent une multiplication des ressources instruments.

Ces mouvements résultent et témoignent à la fois de l'intégration progressive de nouveaux repères didactiques montrant que l'enseignante a acquis une plus grande maîtrise des outils nécessaires à un enseignement de robotique. En voici quelques exemples : d'un point de vue technique, la sensibilité du robot à l'état du sol ; d'un point de vue didactique, l'importance de l'orientation initiale du robot ; d'un point de vue de la pratique informatique, l'intérêt des tâches de test et de débogage ; et d'un point de vue pédagogique, l'intérêt d'interdire une commande lors d'une tâche de codage.

Son action a été fortement soutenue par son expérience, mais aussi contrainte par les ressources disponibles à l'époque de la recherche. Il y a bien des manuels spécialisés dans l'enseignement de l'informatique au primaire, cependant des chercheurs (Vandeveldt *et al.*, 2022) ont montré que les contenus y sont centrés sur l'algorithmique, minorant ce qui relève de la technique et des usages de l'informatique dans la société. Par ailleurs, l'enseignante, qui ne dispose pas de ces manuels en classe faute de budget, se base plutôt sur les manuels de mathématiques de la classe, ou sur des sites spécialisés. (Elle fait par exemple référence à l'ouvrage *1, 2, 3... codez!*<sup>3</sup>)

Les manuels de mathématiques, bien plus usuels pour les enseignants, accordent une place très réduite à la programmation des robots et n'explicitent pas suffisamment les particularités du fonctionnement de ces systèmes. Il est important que les ressources prennent en compte non seulement les aspects informatiques et de programmation, mais aussi les aspects mécaniques et didactiques qui ne sont pas du ressort des seules mathématiques. Il subsiste un écart irréductible entre l'idéalisation mathématique des formes géométriques ou des nombres, et le comportement irrégulier d'un objet technique par nature imparfait.

Nous avons vu que l'utilisation successive de différents repères de déplacement pouvait générer des confusions entre des systèmes de codage basés sur des symboles similaires (des flèches) et incompatibles. C'est en effet parce que le robot est un objet technique avec des roues et des axes de rotation, que son déplacement ne peut pas être réduit à celui d'un point, comme le fait le codage en repère absolu.

Le choix d'inscrire dans les textes officiels des activités de codage et décodage de déplacements en privilégiant le lien avec les

3. <https://fondation-lamap.org/projet/123-codez>. Plutôt que d'un manuel classique, il s'agit d'un recueil d'activités organisées de manière thématique et librement accessible en ligne.

mathématiques apparaît finalement préjudiciable à la bonne compréhension des spécificités de la robotique, de la programmation et à la prise en compte des besoins des élèves pour ces activités.

## Références

Baron, G.-L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie. Recherches en éducation*, vol. 195, n° 2, p. 51-62. <https://doi.org/10.4000/rfp.5032>

Greff, É. (1998). Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants, *Revue Sciences et techniques éducatives*. <https://doi.org/10.3406/stice.1998.1372>

Komis, V., et Misirli, A. (2012). Jeux programmables de type Logo à l'école maternelle, *Adjectif : analyse et recherche sur les TICE*. <http://www.adjectif.net/spip/spip.php?article140&lang=fr>

Nogry, S., et Spach, M. (2022). Robotics learning at elementary school : Constructing abstractions using multiple instruments, *Frontiers in Education Technology*, vol. 5, n° 2, p. 36. <https://doi.org/10.22158/fet.v5n2p36>

Spach, M. (2017). *Activités robotiques à l'école primaire et apprentissage de concepts informatiques : quelle place du scénario pédagogique ? Les limites du co-apprentissage*, thèse de l'université Sorbonne Paris Cité. <https://theses.hal.science/tel-02271924>

Stokes, A., Aurini, J., Rizk, J., Gorbet, R., et McLevey, J. (2023). Using robotics to support the acquisition of STEM and 21st-century competencies : Promising (and practical) directions, *Canadian Journal of Education / Revue canadienne de l'éducation*, vol. 45, n° 4, p. 1141-1170. <https://doi.org/10.53967/cje-rce.5455>

Vandeveldde, I., Fluckiger, C., et Nogry, S. (2022). Resources and textbooks for computer science education in French primary schools, *International Association for Research on Textbooks and Educational Media E-Journal*, vol. 14, n° 1, p. 1-20. <https://doi.org/10.21344/iartem.v14i1.954>

Pour citer ce chapitre :

Kummer-Hannoun, Pascale, et Baron, Georges-Louis (2024). « Quelles ressources pour enseigner l'informatique dans le premier degré ? », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 69-84. <https://doi.org/10.53480/2024iecare04r>

## **PARTIE 2**

# **APPRENDRE L'INFORMATIQUE À L'ÉCOLE**





# Ruptures et continuités dans les représentations de l'informatique et de son apprentissage chez les élèves de cycles 3 et 4

# 5

Katell BELLEGARDE<sup>1</sup>  
Laetitia BOULC'H<sup>2</sup>  
Isabelle VANDELVE<sup>1</sup>

1. Université de Lille, ULR 4354,  
CIREL - Centre Interuniversitaire  
de Recherche en Éducation de Lille,  
59000 Lille, France

2. Université Paris Cité, Laboratoire  
EDA, 75006 Paris, France

Les programmes de 2016 réactualisent la question de l'informatique comme objet d'enseignement et non uniquement comme outil au service d'autres disciplines (Baron et Drot-Delange, 2016). Cette réintroduction des sciences informatiques dans les programmes de L'Éducation nationale à l'école élémentaire et au collège avec l'apparition du nouveau socle commun de compétences (Bulletin officiel, 2015) interroge particulièrement le rôle que joue l'école dans la (trans)formation des rapports à l'informatique des élèves.

Face à ces prescriptions nouvelles, les enseignants, en l'absence de formation spécifique, se retrouvent confrontés à l'enseignement de l'informatique auprès de leurs élèves sans qu'ils ne soient toujours à même d'identifier clairement les contenus et enjeux de savoirs et de maîtriser les nouveaux outils pédagogiques inhérents à cet enseignement (Baron *et al.*, 2015; Fluckiger et Bart, 2012; Spach, 2017; Bellegarde *et al.*, 2021). De leur côté, les élèves font largement usage d'outils informatisés hors des murs de l'école et développent ainsi une culture numérique, pouvant se heurter aux formes scolaires (Lahire, 1993; 2008) et figures légitimes du savoir (Melin, 2019).

Fondé sur des entretiens et des enquêtes par questionnaire menées auprès d'élèves de 9 à 14 ans, ce travail se propose d'identifier et de comprendre les ruptures et continuités dans les rapports à l'informatique de la fin de l'école élémentaire et du collège. Il permet de montrer comment les représentations de l'informatique et de son apprentissage se construisent et se transforment au cours de leur parcours personnel, en référence à leur culture enfantine en constante évolution.

## Cadre théorique : les rapports à l'informatique d'élèves de cycles 3 et 4

Cette première partie rend compte des investigations théoriques menées en vue d'approcher la question des rapports à l'informatique d'élèves de cycle 3. Dans un premier temps, nous reviendrons sur la question de l'enseignement de l'informatique au sein de l'école (désignée, communément par l'appellation « éducation au numérique »). Dans un second temps, nous nous attacherons à appréhender le concept de rapports à l'informatique.

### Informatique scolaire : les enjeux de savoir en question

En France, l'informatique n'a pas le statut de discipline scolaire ; les contenus d'enseignement sont abordés de façon transversale dans différentes disciplines. Ce manque de légitimité a conduit à « une mise en sommeil [...] de la didactique de l'informatique » (Fluckiger, 2019). Cette situation s'explique notamment par la difficulté à s'accorder sur une ontologie de l'informatique : cette dernière relève (1) de la science et de manières de penser spécifiques (Wing, 2008), (2) de la technologie, dans le sens de l'étude des techniques et outils, et (3) d'usages spécifiques, de l'utilisation d'outils informatisés (Bruillard, 2009b ; Fluckiger, 2019). À l'instar de Fluckiger (2019), nous pensons que la didactique de l'informatique pourrait s'intéresser à l'ensemble de ces manifestations et donc à des contenus et savoirs ne relevant pas uniquement du domaine scientifique, et ceci, même si cette pluralité d'objets ne contribue pas à l'unité et la visibilité de ce domaine de recherche. Cette triple vision de l'informatique se retrouve en filigrane dans les programmes et manuels scolaires (cf. le chapitre 1 « Quel enseignement de l'informatique à l'école primaire en France ? Réflexions sur 40 ans de développements ») : les élèves doivent à la fois apprendre les bases de la programmation (par exemple, en lisant un message codé, composé de flèches afin de tracer sur un damier le parcours d'un robot virtuel), le fonctionnement des outils informatisés (décrire l'architecture simple d'un dispositif informatique, montrer comment des périphériques sont reliés à l'unité centrale, etc.) et leur maîtrise (s'approprier un logiciel de traitement de texte, apprendre à écrire avec un clavier, etc.).

Les difficultés que pose la désignation de ces contenus d'enseignement s'observent à travers le glissement sémantique opéré entre informatique et numérique : on note ainsi un emploi croissant du terme *numérique* dans le champ scolaire au détriment de celui d'*informatique* (Fluckiger, 2019). Le terme numérique désigne la manière dont l'information est traitée et stockée, sous la forme

de nombres. En contexte scolaire il renvoie « à tout ce qui est lié à l'instrumentation des activités et des pratiques par des systèmes de traitement de l'information » (Baron et Boulc'h, 2011) et conduit à inscrire ce champ de pratique dans une nébuleuse : « Au lieu de se limiter à ce qui est relatif aux nombres, il a été utilisé comme un équivalent et souvent comme une euphémisation de ce qu'on reliait autrefois à l'informatique et aux logiciels. On le trouve notamment en relation avec l'école et la pédagogie sans que la signification de ces assemblages soit toujours claire » (*ibid.*). Même si, en France, l'éducation au numérique est plus largement évoquée que l'éducation à l'informatique, « l'informatique [n'en est pas moins] [. . .] la science fondamentale de notre monde numérique », ce qui permet d'expliquer ce monde et de le rendre possible (Société informatique de France – SIF, 2014). Dans le cadre de cette contribution, nous préférons donc faire usage du terme *informatique* qui permet d'écarter de notre démarche scientifique, le numérique, prénotation (au sens de Durkheim, 1987) qui, insérée dans le langage courant, est empreinte de confusion et mélange indistinctement « de(s) impressions vagues, de(s) préjugés, [. . .] de(s) passions » (*ibid.*, p. 27) et des mythes relatifs à l'apprentissage et au numérique (Amadiou et Tricot, 2014).

### **Culture numérique infantile et forme scolaire : du « rapport à » aux représentations de l'informatique**

Les enfants d'aujourd'hui, natifs du numérique, grandissent dans un environnement où les technologies informatisées sont omniprésentes. Aussi, les études quantitatives montrent un taux d'équipement et d'usage important et grandissant chez les enfants et adolescents, avec une progression significative avec l'âge (Ipsos, 2017). On observe, par exemple, un temps de connectivité à Internet presque deux fois supérieur entre les adolescents de quinze-seize ans et les enfants de neuf et dix ans (164 minutes par jour contre 71) (Blaya et Alava, 2012). Dix-onze ans constitue une période charnière de leurs pratiques des technologies de l'information et de la communication (TIC), avec l'arrivée du premier smartphone, et une diversification des usages (Hadopi, 2017). Selon la maxime « tout, partout et tout le temps » (p. 2), les biens culturels dématérialisés sont consommés par les enfants (8-14 ans), chaque jour, en abondance et tout au long de la journée (hors temps scolaire) via divers appareils (smartphone, tablette, console, etc.).

Dès leur plus jeune âge, les enfants développent ainsi des pratiques ordinaires du numérique, une culture numérique définie comme « l'ensemble de valeurs, de connaissances et de pratiques qui impliquent l'usage d'outils informatisés, notamment les pratiques de

consommation médiatique et culturelle, de communication et d'expression de soi » (Fluckiger, 2008, p. 51). Cette culture numérique s'inscrit dans une culture enfantine singulière, c'est-à-dire, dans un « ensemble de pratiques, de connaissances, de compétences et de comportements qu'un enfant doit connaître et maîtriser pour intégrer le groupe de pair » (Delalande, 2003). Construite en étroite relation avec la maturité physique et intellectuelle de l'enfant, elle est évolutive : un enfant développe de nouvelles préoccupations et représentations de son environnement, de nouvelles pratiques au fur à mesure qu'il grandit. En cela, elle participe à la construction d'une identité commune chez les enfants d'une même tranche d'âge. Ceux-ci produisent leur propre culture en se référant au monde des adultes, en se nourrissant d'une culture globale à laquelle ils participent eux-mêmes (Delalande, 2003 ; Arleo et Delalande, 2010).

Cette culture numérique enfantine se construit, le plus souvent, hors des temps scolaires. Les modes d'appropriation des technologies informatisées se distinguent des modes d'apprentissage en vigueur à l'école et, ceci, même si l'institution scolaire entend jouer un rôle dans l'éducation au numérique des élèves. Comment s'opère la rencontre entre la culture numérique enfantine et la forme scolaire qui constitue le cadre légitime dans lequel l'école entend former les élèves ?

La forme scolaire organise de manière spécifique la relation d'enseignement-apprentissage, en la différenciant notamment de ses modes informels (dans la famille, par tâtonnement, par mimesis etc.) (Reuter, 2007). Dans ces formes sociales d'apprentissage, l'appropriation des savoirs passe par un ensemble de pratiques et d'exercices qui doivent inscrire l'individu dans une raison scolaire, un rapport scriptural-scolaire au langage et au monde : « C'est tout un rapport au langage et au monde que les pédagogues entendent inculquer aux élèves à travers les multiples pratiques langagières (orales ou écrites) engendrées dans des formes sociales scripturales scolaires : une maîtrise symbolique, seconde, qui vient ordonner et raisonner ce qui vient de la simple habitude, du simple usage, de la pratique sans principe explicite » (Lahire, 1993). La culture scolaire pourrait ainsi entrer en conflit avec la culture numérique enfantine, cette dernière se trouvant alors transformée : le savoir est objectivé et non incorporé, les modèles d'usage sont explicites et non appris par mimesis ; l'apprentissage se fait à l'école et non dans la pratique.

La notion de « rapport à » constitue un cadre heuristique pour penser ce conflit de normes, entre raison scolaire et raison enfantine. Beillerot (1989) considère cette notion comme centrale pour interroger l'appropriation du savoir chez l'apprenant. Elle désigne un ensemble de rapports qui « viennent à un sujet d'une façon directe et indirecte, de sa famille, de son milieu social, de la

période historique, des apprentissages divers jusqu'au premier temps du lait et du sein » (p. 175). Le « rapport à » envisage une configuration particulière entre un individu et un objet, configuration constituée d'une pluralité d'éléments. Elle s'oppose de façon radicale à une posture déterministe qui réduirait les trajectoires individuelles à la simple appartenance à une classe sociale pour se situer dans une démarche compréhensive du sujet pris dans sa singularité. Charlot (1997) à travers la notion de « rapport au savoir » a tout particulièrement mis en évidence l'intérêt d'adopter une lecture positive de l'échec scolaire qui s'attache à l'expérience singulière des élèves, à leur interprétation du monde, à leurs activités. L'introduction de la notion de « rapport à » dans les sciences de l'éducation marque ce qui a pu être décrit comme une rupture épistémologique puisque l'échec scolaire est appréhendé en termes de processus pluriels en interaction (Bautier, 2002)<sup>1</sup>. Nous proposons d'envisager les rapports à l'informatique comme « une relation de sens et de valeur attribuée à des activités, à des situations, à des produits (institutionnels, culturels, sociaux) liés [à l'informatique], relation construite par le sujet à propos de [l'informatique], de ses usages et de son apprentissage »<sup>2</sup>.

Dans la lignée des travaux de Bellegarde (2015), il est possible de catégoriser les rapports à l'informatique en trois dimensions : (1) pratiques et usages situés de l'informatique, (2) représentations de l'informatique et de son apprentissage<sup>3</sup> et (3) fonctionnements et stratégies dans les pratiques de l'informatique. Dans le cadre restreint de cette contribution, nous nous intéressons particulièrement à la seconde dimension et nous proposons de décrire et d'analyser les représentations des élèves d'élémentaire et de collègue concernant l'informatique et son apprentissage : comment se forment-elles en lien avec leurs expériences personnelles et scolaires ? Quelles ruptures et continuités dans ces représentations peut-on observer chez ces élèves d'âges différents ?

Nos investigations théoriques mettent en avant la place ambivalente qu'occupe l'informatique au sein du système scolaire. Malgré la réintroduction de l'apprentissage de l'informatique dans les programmes scolaires, cette dernière conserve le rôle d'outil au service d'autres apprentissages / disciplines dans une perspective transversale. De même, on observe un manque de formation des enseignants qui, conjugué au glissement sémantique opéré entre informatique et numérique, accentue le caractère nébuleux de l'informatique scolaire et la difficulté à identifier les concepts et contenus disciplinaires sous-jacents. Dans ce contexte, comment les élèves définissent-ils l'informatique ? En ont-ils une vision élargie (usages, sciences, technologies – Fluckiger, 2019) ou restreinte ? Dans quelles mesures ces conceptions donnent-elles sens à l'apprentissage de l'informatique ?

1. Bautier (2002) envisage cette rupture épistémologique à travers la notion de « rapport au langage ».

2. Cette définition s'appuie sur celle formulée par Leclercq et Bellegarde (2015) au sujet des rapports à l'écrit, celui-ci constituant, tout comme l'informatique, une pratique sociale.

3. Nous définissons les représentations comme un ensemble de valeurs, d'opinions, d'informations, et de croyances relatives à l'objet de la représentation (Jodelet, 2003). Construites et partagées par les membres d'un groupe social (Moscovici, 2003), elles sont une manière de penser, de s'approprier, d'interpréter notre réalité quotidienne et notre rapport au monde. Les objets de notre environnement quotidien, y compris les disciplines scolaires, font l'objet de représentations (Abric, 2003).

Nous avons également souligné le développement d'une culture numérique précoce qui s'inscrit dans une culture enfantine singulière et évolutive. Baignant dès le plus jeune âge dans un environnement informatisé, on observe avec l'avancée en âge, une multiplication et une diversification des pratiques ordinaires. Comment la culture numérique des élèves évolue-t-elle ? Quelles différences notables observe-t-on entre les représentations de l'informatique et de son apprentissage des collégiens et celles des élèves de primaire ?

Enfin, rappelons que cette culture numérique enfantine s'est construite le plus souvent en dehors des temps scolaires, dans la pratique quotidienne, sans un temps formel d'apprentissage. Cette construction se distingue des modes d'apprentissage en vigueur à l'école. Comment s'opère alors la rencontre entre la culture numérique enfantine et la forme scolaire qui constitue le cadre légitime dans lequel l'école entend former les élèves ? Comment l'école participe-t-elle à la (trans)formation des rapports à l'informatique des élèves ?

Ces considérations ont construit le cadre de nos études respectives portant sur les rapports à l'informatique d'élèves de cycle 3 et 4 (milieu et fin d'école primaire et collège).

Les données présentées dans ce chapitre sont issues d'enquêtes de terrain distinctes mobilisant des méthodologies qualitatives et quantitatives complémentaires. Elles ont en commun d'approcher la question des rapports à l'informatique chez des élèves de la fin de l'école élémentaire et du collège.

## **Méthodologie**

Ce travail se situe dans une approche compréhensive, qui, d'après le principe d'intercompréhension, admet que l'humain peut saisir le vécu, l'expérience d'un autre et ainsi avoir accès au sens qu'il donne à la réalité étudiée, ici les rapports à l'informatique (Paillé et Mucchielli, 2003). Deux modalités d'élaboration du corpus ont été retenues : des entretiens et des questionnaires. Les données ont été recueillies en 2020–2021 soit cinq ans environ après la réintroduction des sciences informatiques dans les programmes du primaire et du collège.

Les entretiens individuels, menés auprès d'élèves des cycles 3 et 4, favorisaient une liberté d'expression chez les élèves, leur permettant de partager leurs points de vue et leurs vécus, avec autant d'authenticité et de profondeur que possible. Seize élèves de primaire et douze collégiens ont participé à des entretiens semi-directifs individuels d'une durée variant de 30 à 60 minutes.

En complément des entretiens menés dans le second degré, des questionnaires ont également été diffusés afin de collecter des données sur un échantillon d'élèves plus large et représentatif. Ils comportaient une quinzaine de questions permettant de recueillir des données sur leurs pratiques de l'informatique à l'école et à la maison ainsi que sur leur attitude envers l'informatique (via des questions issues du questionnaire de Moskal *et al.*, 2006).

Ayant conscience des effets du contexte scolaire sur les réponses recueillies, nous avons veillé à instaurer un climat de confiance en explicitant les objectifs de notre recherche (« comprendre leurs pratiques de l'informatique pour réfléchir à la manière d'enseigner cette discipline ») et en valorisant l'importance de leurs témoignages accueillis sans jugement.

### **Population**

Les enquêtes menées auprès d'élèves de cycle 3 ont concerné quatre écoles situées dans le bassin minier du nord de la France. Deux d'entre elles appartiennent au réseau d'éducation prioritaire. Sur les seize élèves de primaire ayant participé au recueil de données, quatre étaient de niveau CM1 et quatorze de niveau CM2.

Au collège, une enquête par questionnaire a concerné 427 élèves (110 élèves de 6<sup>e</sup>, 89 élèves de 5<sup>e</sup>, 176 élèves de 4<sup>e</sup> et 52 élèves de 3<sup>e</sup>) répartis dans cinq établissements privés ou publics de l'ouest et du nord de la France. Les entretiens, quant à eux, ont concerné six collégiennes et collégiens de 5<sup>e</sup> et six collégiennes et collégiens de 3<sup>e</sup> dont la moitié s'était inscrite volontairement dans un club ou des ateliers robotique/scratch offerts par la commune ou proposés au sein du collège, en dehors des temps d'enseignements obligatoires.

### **Modalités de traitement des données**

Les réponses des élèves et des collégiens ont été analysées selon les mêmes grilles d'analyse et classées selon les mêmes catégories. Les analyses réalisées ont porté sur deux dimensions : les représentations de l'informatique des élèves de primaire et collégiens et les représentations de l'apprentissage de l'informatique.

Pour la première dimension, les réponses de chaque élève ont été classées au sein de catégories élaborées à partir des définitions de l'informatique de Bruillard (2009b) et Fluckiger (2019). Pour la seconde dimension, nous nous intéressons, dans un premier temps, à ce que les élèves et les collégiens pensent avoir appris en informatique dans un cadre scolaire. Les réponses recueillies dans les questionnaires et lors des entretiens ont été classées en plusieurs catégories construites à partir des travaux de Charlot (1997, 1999).



Dans un second temps, nous avons souhaité comprendre à travers cette dimension, les finalités, le sens que donnaient les élèves au fait d'apprendre l'informatique à l'école et au collège. À partir des travaux de Carré (1998) et de Carré et Caspar (2017), portant sur l'engagement d'adultes en formation, nous avons identifié quatre motifs donnés par les élèves témoignant des finalités de l'informatique scolaire. Les explications précises concernant la constitution des différentes catégories seront fournies dans la troisième partie.

## **Résultats : l'informatique et son apprentissage, des représentations inscrites dans une culture enfantine**

Cette troisième partie dédiée aux résultats de nos enquêtes interroge les représentations de l'informatique et de son apprentissage que se sont forgés des élèves de la fin de l'école élémentaire (CM1 et CM2) et du collège au cours de leur parcours personnel. Elle propose spécifiquement de saisir, d'une part, les manières dont ces élèves définissent ce qu'est pour eux l'informatique, et d'autre part, comment ces mêmes élèves envisagent l'informatique en tant qu'objet scolaire, le sens et la finalité qu'ils donnent à cet apprentissage.

Nos résultats montrent, chez des élèves âgés entre 9 et 15 ans, une tendance à percevoir l'informatique à travers le prisme de leurs pratiques ordinaires, pratiques inscrites dans une culture enfantine évolutive. Ils confortent les travaux d'Abric (2003), à savoir l'influence des pratiques des sujets sur les représentations sociales, ici, celles de l'informatique. La culture numérique enfantine (Fluckiger, 2008) prend ainsi des configurations originales en fonction de l'âge des élèves : la valeur donnée à l'outil informatique est exponentielle de l'âge de 9 ans à celui de 15 ans tandis que celle donnée à son apprentissage est déclinante pour cette même tranche d'âge.

### **L'informatique, un outil culturel ancré dans des pratiques ordinaires**

Lors de nos entretiens avec les élèves, nous avons fait le choix de les interroger, dès le début, sur ce qu'est l'informatique pour eux. Il s'agissait ainsi de partir de leurs propres représentations sans induire par nos questionnements une représentation construite au fur et à mesure de l'échange. À partir des définitions de l'informatique de Bruillard (2009b) et Fluckiger (2019), les réponses des élèves ont été classées en trois catégories : (1) comme une science :

l'élève fait référence à des notions précises comme « algorithmique », « pensée informatique », « programmation », « boucle » ou évoque le concept de programmation ; (2) en tant que technologie : l'élève fait référence à l'étude des techniques et des outils ainsi qu'aux relations qu'ils entretiennent entre eux (mise en réseau) ou qui existent entre les différents composants (câbles, wifi. . .) ; (3) par ses usages : sont distinguées ici trois sous-catégories : (3a) les usages sociaux qui renvoient aux pratiques sociales ordinaires en dehors du cadre scolaire ; (3b) les usages scolaires où l'informatique est perçue comme un outil utilisé en classe pour les apprentissages ; (3c) les usages d'outils informatisés lorsque l'élève restreint l'informatique aux actions, au faire (il cite un outil, une activité réalisée sur un logiciel, etc.). Un même élève pouvant fournir des réponses variées, il est possible que toutes ou plusieurs de ces différentes catégories soient représentées dans les propos d'un seul répondant.

Les résultats de nos enquêtes soulignent l'inscription de l'informatique dans une culture enfantine qui tend à se construire progressivement, notamment, à partir de cet objet/outil pour lequel les élèves de primaire et de collège interrogés ont développé des représentations en lien avec leurs pratiques et parcours personnels. Les réponses des élèves de primaire montrent majoritairement des représentations pragmatiques, dans le sens où l'informatique est définie surtout sous le prisme des usages, « c'est pour faire » (16 élèves), et dans une moindre proportion comme une technologie (trois élèves) ou une science (un élève). Chez les collégiens, les réponses données démontrent une plus grande diversité dans leurs représentations de l'informatique : on retrouve toujours une forte présence des usages mais elle est aussi considérée comme une science par un tiers des élèves interrogés. Tout comme en primaire, la dimension technologique n'est que peu mise en avant.

### **L'informatique sous le prisme des usages : « c'est pour faire. . . »**

Pour les élèves de primaire et de collège, l'informatique est d'abord synonyme d'outils informatisés (neuf élèves et cinq collégiens) sans référence à leurs dimensions techniques et scientifiques ; elle renvoie largement aux « écrans », à l'ordinateur, aux logiciels et applications utilisés. De même, la définition de l'informatique donnée dans les questionnaires par les collégiens consiste souvent en un simple *listing* (liste) d'outils : « ordinateurs », « téléphone portable », « tablettes », « appli », « robot », « carte micro-bit ». Ces réponses *listings* sont observées dans 42 % des questionnaires, constat qui s'observe aussi dans la moitié des entretiens menés avec des collégiens.

Informatique, c'est genre Facebook, Google, les réseaux, en fait. (Timéo, CM2)

Je pense que l'informatique pour moi c'est les écrans, les claviers, les ordinateurs et tout ce qui est logiciel et application aussi. Jouer aux jeux vidéo peut-être c'est de l'informatique aussi ? Y a un écran, des manettes. . . alors on peut dire que les consoles c'est aussi l'informatique. (Manelle, 5<sup>e</sup>)

Les élèves de primaire ont également largement défini l'informatique au travers des usages sociaux qu'ils développent dans leur contexte quotidien (huit élèves). Les outils informatisés constituent alors, pour sept élèves, une source de distraction, un moyen pour ne pas s'ennuyer facile d'accès et offrant une pluralité de possibilités : jouer à des jeux, regarder des vidéos sur YouTube, discuter avec ses amis sur les réseaux sociaux, etc. L'informatique est également définie comme une source d'apprentissage par six élèves de primaire, apprentissage qui s'opérerait via divers logiciels conçus à cet effet ou via la recherche d'information sur des moteurs de recherche. L'aspect ludique apparaît nettement dans le discours des élèves ; il s'agit de s'occuper et d'apprendre tout en s'amusant.

Si tu devais expliquer à un enfant qui ne sait pas du tout ce que c'est l'informatique, qu'est-ce que tu lui dirais ?

Je lui dirais qu'on peut apprendre des choses, qu'il peut jouer [ . . . ]. Je lui montrerais des jeux d'exercices, c'est une méthode d'apprentissage mais sans travailler forcément. [ . . . ] L'apprentissage par le jeu, [ . . . ] c'est bien car il y a des gens s'ils veulent apprendre des choses, [ . . . ] ils ne sont pas obligés de se plonger dans des livres pendant des heures, ils peuvent aller sur Internet et jouer à des jeux en même temps que travailler, ça les stimule. (Titouan, CM2)

Cette tendance à définir uniquement l'informatique sous le prisme des usages notamment sociaux est également très présente chez les collégiens. Ainsi dans 93 % des questionnaires, au moins un usage est cité : « jouer aux jeux vidéos », « aller sur Internet », « envoyer un MMS », « liker et commenter une photo sur Insta », « installer et se servir des applications » « poster une vidéo TikTok », « créer des chaînes YouTube », « faire du Word », « créer un jeu vidéo sur Scratch ».

Si l'informatique est définie pour une large partie des élèves à travers leurs usages, que ce soit sous le prisme des outils informatisés ou des pratiques sociales, seuls quelques-uns ont envisagé l'informatique comme un outil pour l'école (trois élèves en primaire, trois collégiens interviewés et 37 % des collégiens ayant répondu au questionnaire).

Quand on ne comprend pas quelque chose ou qu'on doit faire un exposé, par exemple, le professeur nous dit qu'on peut aller faire des recherches et les noter pour faire notre exposé, par exemple sur l'ordinateur il y a Internet et c'est ce qui nous permet de trouver des choses qu'on ne trouve pas dans le dictionnaire ou dans le Bescherelle ou tout ce qui s'en suit. (Célinie, CM2)

C'est tout ce que tu peux faire avec l'ordinateur. Par exemple en classe on fait de l'informatique, on doit faire des exposés, on doit aller chercher sur Internet et après on projette avec le tableau numérique interactif (TNI) les informations qu'on a trouvées pour l'exposé. (Gabriel, 5<sup>e</sup>)

### **L'informatique en tant que technologie ou science**

Les représentations de l'informatique en tant que technologie ou science se retrouvent de façon plus marginale dans les propos des élèves de primaire et des collégiens que celles qui concernent les usages. On observe cependant un net décalage entre les réponses données par les répondants en fonction de leur âge.

Ainsi, chez les élèves de primaire, les conceptions de l'informatique s'ancrent dans une culture numérique enfantine, dans des modes d'appropriation informels pour lesquels l'informatique en tant que science et technologie du monde numérique (SIF, 2014) n'a que peu de raison d'être. L'école primaire semble peu impacter les grilles de lecture de l'informatique que se sont construits ces élèves au cours de leur parcours personnel. Aussi, seuls trois élèves de primaire, dans leur définition de l'informatique, ont envisagé celle-ci comme une science et/ou une technologie. Même si ces représentations sont encore fragiles, confuses, inexactes, elles montrent néanmoins des réflexions et questionnements sur ce qu'est l'informatique et sur la manière dont fonctionnent les outils informatisés.

En fait tant que ça touche à tout ce qui est scientifique, j'adore. [...] C'est scientifique oui, parce que c'est l'humain qui a créé ça et ça a été fait avec des études mais ils ont dû passer beaucoup de temps dessus. [...] Je pense qu'ils ont dû fabriquer par leurs propres moyens, ils ont dû étudier comment faire et ils se sont dit « ça je vais l'appeler comme ça » et dès qu'il y a eu ça, c'était une révolution, c'était la révolution de l'informatique, Internet. (Titouan, CM2)

La représentation de l'informatique comme « science » est davantage présente chez les collégiens ( $n = 4$ ).

L'informatique, ça concerne tout ce qui est programmation, le langage pour programmer des ordinateurs, pour programmer des robots. Donc, c'est pas juste savoir utiliser un ordinateur mais aussi comment il fonctionne, pourquoi il fonctionne pas et lui dire quoi faire. (Théophile, 3<sup>e</sup>)

De même, dans les questionnaires, 38 % des collégiens (essentiellement les 3<sup>e</sup> et 4<sup>e</sup>) mettent au moins une fois en avant des dimensions scientifiques. Ils utilisent des termes comme : « code/r », « pour la programmation des ordinateurs et des robots », « faire du Python », « les boucles », « les algorithmes ». La dimension scientifique est également bien présente dans cinq des douze interviews réalisées.

Notons toutefois, que si les savoirs et les connaissances techniques semblent plus accessibles aux collégiens, les connaissances sont très imprécises et l'informatique reste encore perçue au travers du prisme des usages sociaux. Ainsi, lorsqu'il a été proposé aux collégiens d'indiquer, parmi une liste d'actions, celles qu'ils considéraient comme étant de l'informatique, les réponses données ont montré une méconnaissance des termes techniques (cf. tableau 1). Ainsi, « Créer un algorithme » est considéré par seulement 8 % des élèves de 6<sup>e</sup>/5<sup>e</sup> et 16 % des élèves de 4<sup>e</sup>/3<sup>e</sup> comme une compétence informatique. Pour eux, ce terme fait référence à une situation fréquente en primaire que les enseignants nomment un algorithme et qui consiste à repérer une répétition dans une suite logique de formes ou d'objets : « Quand on fait une répétition, une sorte de suite logique » (Serge, 3<sup>e</sup>).

De même, 47 % des collégiens ont répondu que « comprendre le fonctionnement d'Internet » ne correspondait pas à de l'informatique. Comme souligné plus haut, l'informatique est surtout définie au travers des usages. Le savoir, dissocié du savoir-faire est considéré, pour beaucoup, comme secondaire et non-essentiel :

On peut faire de l'informatique sans comprendre le fonctionnement d'Internet, les réseaux, les câbles et tout. Si tu sais utiliser Google, si tu sais faire une recherche dans le moteur de recherche, après tu tombes sur ce que tu cherches, les sites et voilà. Y a pas besoin de connaître le fonctionnement technique d'Internet pour être efficace sur Internet et trouver ce que tu veux. Et puis, si l'informatique, c'est programmer des machines, des robots, ben on peut le faire sans Internet et sans comprendre Internet, il faut juste un ordi et une appli. (Clara, 5<sup>e</sup>)

**TABEAU 1** – Répartition des réponses à la question « Parmi ces actions, coche celle(s) qui, selon toi, correspond(ent) à de l'informatique ».

	Total (n = 447)	6 <sup>e</sup> /5 <sup>e</sup> (n = 209)	4 <sup>e</sup> /3 <sup>e</sup> (n = 238)
Envoyer un email	84	90	79
Saisir au clavier	69	77	62
Créer un algorithme	12	8	16
Comprendre le fonctionnement d'Internet	47	33	59
Télécharger un fichier	88	85	91
Faire une recherche sur Internet	87	92	82
Programmer un robot	92	93	91

Ces tendances concernant les représentations de l'informatique ont également été observées dans une étude complémentaire menée au sein d'une classe de CM1-CM2, faisant appel au concept de classes de situation (Vergnaud, 1989). Nous n'en détaillerons pas, ici, tous les résultats obtenus, mais un résumé des éléments saillants est disponible en annexe.

### **Apprendre l'informatique à l'école, une pratique scolaire opaque ancrée dans l'action**

Les élèves du premier et second degré ont été questionnés sur ce qu'ils pensaient avoir appris en informatique à l'école ou au collège puis sur l'intérêt de suivre un enseignement de l'informatique à l'école.

Les réponses ont d'abord été classées selon trois catégories construites à partir des travaux de Charlot (1997, 1999) : (1) l'appropriation d'un savoir objet qui peut être évoqué sans référence à la situation dans laquelle il a été appris, (2) la capacité à réaliser une action ou à maîtriser une procédure mais sans possibilité de se détacher de la situation dans laquelle elle a été abordée, (3) aucun apprentissage verbalisé.

Puis, les finalités de l'apprentissage de l'informatique, exprimées par les élèves, ont été classées en quatre motifs (Carré, 1998) : motif vocationnel, motif opératoire de type personnel, motif opératoire de type scolaire et motif hédonique.

L'informatique pratiquée dans un cadre scolaire apparaît comme un objet opaque, que les élèves utilisent sans qu'ils ne puissent toujours identifier les savoirs et compétences sous-jacentes ainsi que les finalités liées à ces apprentissages.

### **Apprendre l'informatique, réaliser/maîtriser un ensemble d'actes *versus* s'approprier un savoir-objet**

L'ensemble des élèves de primaire et des collégiens interrogés entretiennent un rapport épistémique au savoir (Charlot, 1997) informatique en lien étroit avec l'action. Ce rapport pratique au savoir prend deux formes : apprendre est d'abord synonyme d'actions réalisées dans une situation donnée d'apprentissage; apprendre a également pour visée la maîtrise d'activités concrètes inscrites dans le contexte social et scolaire des élèves.

Dans le premier cas, il dénote la non-adoption d'une posture de secondarisation (Bautier et Goigoux, 2004) puisque leurs difficultés résident dans l'identification des enjeux cognitifs des tâches en informatique réalisées à l'école. C'est le cas pour quatorze élèves de primaire et cinq collégiens interrogés sur ce qu'ils ont appris à l'école. Ceux-ci restent alors « enfermés (imbriqués) dans la matérialité de l'opération sans accéder à son sens » (Charlot, 2001). Ce qui est appris n'est pas un objet qui peut être évoqué de façon réflexive sans référence à un acte ou un ensemble d'actes en situation. Cette difficulté à énoncer ce qu'ils ont appris et travaillé à l'école est perceptible au regard de l'aspect succinct, souvent embrouillé, de leur discours sur l'informatique voire de l'impossibilité d'en parler.

Il y avait une lettre [qui s'affichait] sur l'écran et il fallait la taper [le plus vite possible] mais c'était trop compliqué (rire) [Qu'est-ce que tu as appris en faisant cette activité?] Je ne sais pas du tout. (Roxane, 10 ans)

Qu'est-ce que tu as appris quand tu as travaillé avec le robot?

J'apprends en faisant des épreuves. [. . .] À faire déplacer le robot.

Qu'est-ce que tu as appris en déplaçant le robot?

Je ne sais pas. (Yanis, 9 ans)

J'ai appris à utiliser Scratch, les blocs, les déplacements des lutins avec X/Y et mettre le son, on peut enregistrer ou utiliser un son qui est déjà dans Scratch. (Emma, 5<sup>e</sup>)

Je saurais même pas dire, pas grand-chose sûrement. (Clara, 5<sup>e</sup>)

Dans le second cas, le rapport pratique au savoir est porteur de sens envers l'apprentissage de l'informatique à l'école. L'acquisition de compétences en informatique est perçue par l'ensemble des élèves de primaire et par cinq collégiens interrogés comme une réponse à leurs besoins quotidiens, comme une aide à la maîtrise de leur environnement social et/ou scolaire. Apprendre l'informatique, c'est alors se rendre capable de maîtriser une activité où l'informatique entre en jeu :

[J'ai appris à] passer des lignes, revenir à la ligne tout ça. Et pour enregistrer, pour photocopier, imprimer. [...] Pour l'instant, c'était en début d'année, après, je pense qu'on va apprendre d'autres choses. (Axelle, 10 ans)

J'ai appris à faire contrôle C, contrôle V ça c'est pour faire copier-coller. C'est pratique, j'avais jamais vu ça avant. Aussi changer la police, la couleur, la forme. Pour écrire un exposé et faire un dossier, c'est important. (Manelle, 5<sup>e</sup>)

Chez trois collégiens, on observe néanmoins une appropriation de savoirs scientifiques qu'ils sont capables de nommer sans faire systématiquement référence à l'activité précise dans laquelle ils ont été abordés<sup>4</sup>. Dans ce cas, les propos tenus, restent cependant très peu élaborés :

On a fait des trucs en maths par rapport à l'algorithmique [...], les pixels 1 et 0. (Serge, 3<sup>e</sup>)

### **Apprendre l'informatique à l'école, des motifs extrinsèques prédominants**

Nous avons souhaité comprendre le sens que donnaient les élèves de la fin de l'école élémentaire et de collège à l'apprentissage de l'informatique à l'école au regard des finalités qu'ils y donnaient. Tous les collégiens et quatorze des seize élèves de primaire interrogés, sont parvenus à verbaliser les raisons, motifs (Carré, 1998) qu'ils percevaient dans cet enseignement scolaire, à identifier un intérêt, pour eux, à court ou moyen terme. Les motifs extrinsèques concernent une majorité d'élèves (neuf élèves et tous les collégiens) pour lesquels cet apprentissage prend sens en dehors des séances en informatique elles-mêmes, celles-ci constituant un moyen pour atteindre des objectifs extérieurs à cet enseignement : poursuivre sa scolarité, effectuer des tâches mettant en jeu l'informatique dans le contexte quotidien ou professionnel, par exemple. Des motifs intrinsèques ont également été verbalisés par sept élèves de primaire ; pour eux, apprendre l'informatique est porteur de valeur en soi et est relatif à l'acte d'apprendre lui-même. Pour les

4. À partir du cycle 4, les notions abordées en informatique sont censées être plus poussées. À la question « fais-tu de l'informatique en classe et à quelle fréquence », 55 % des 4<sup>e</sup> et 68 % des 3<sup>e</sup> déclarent en faire une fois par semaine ou plus. Les cours d'informatique seraient plutôt l'apanage des enseignants de mathématiques et surtout de technologies. L'informatique en cours de mathématiques est surtout citée par les élèves les plus âgés qui précisent y apprendre des bases de programmation et d'algorithmique via des activités le plus souvent débranchées. Les activités citées en technologies sont plus variées, il peut aussi bien s'agir de la familiarisation avec l'outil ordinateur (savoir enregistrer un fichier, réaliser une recherche Internet en vérifiant ses sources), que de l'utilisation de logiciel et de robot pour la programmation.



motifs extrinsèques, le sens donné à l'informatique scolaire est orienté vers l'apprentissage, l'acquisition de compétences en informatique ; pour les motifs intrinsèques, ce sens se situe dans la participation même à des séances d'apprentissage mettant en jeu l'informatique. Notons que pour un même élève plusieurs motifs, qu'ils soient extrinsèques ou intrinsèques, ont pu être énoncés ; ainsi, pour un même élève ceux-ci peuvent se combiner.

Concernant tout d'abord les motifs extrinsèques, trois types ont été identifiés chez les élèves : il s'agit, par ordre de représentativité dans leur discours, des motifs vocationnels (sept élèves de primaire et six collégiens), opératoires personnels (quatre élèves et quatre collégiens) et opératoires scolaires (trois élèves et cinq collégiens).

Les motifs vocationnels renvoient à l'acquisition de compétences en informatique pour le travail plus tard, sans que les élèves de primaire ne sachent souvent identifier explicitement les tâches que pourrait recouvrir la pratique de l'informatique en milieu professionnel. D'ailleurs, sur les sept élèves se situant dans ce type de motifs, quatre ont lié directement l'acquisition de compétences en informatique au fait de devenir informaticien, ne percevant pas le transfert de ces compétences à d'autres métiers :

Je pense que c'est très bien. [...] Peu importe notre travail, ça pourra toujours nous aider. [...] Par exemple, je travaille dans une grande entreprise où la principale source de recherche, c'est les ordinateurs, je serai bien entraîné. [...] La source principale de recherche pour rechercher des choses, les passer à nos collègues etc. [...] Si je travaille dans une grande entreprise, la source principale, c'est les ordinateurs, j'irai très très vite. (Titouan, CM2)

C'est important parce que si on [...] veut devenir informaticien, il faut savoir faire, sinon c'est bien. [Est-ce que c'est important si on ne veut pas devenir informaticien ?] Oui [...], comme ça, si on a un frère ou une sœur qui ne sait pas utiliser l'ordinateur, on peut lui expliquer. (Leila, CM2)

Le motif vocationnel est évoqué de façon plus large au collège, les élèves percevant que « l'ordinateur », « l'intelligence artificielle », ou encore « les robots » ne sont pas spécifiques à un métier en particulier et que l'apprentissage de l'informatique dans le secondaire est donc une condition *sine qua none* à toute réussite professionnelle :

C'est indispensable, aujourd'hui, il y a de l'informatique partout. Oui c'est sûr parce que maintenant tout

se fait sur ordinateur et si tu veux pas être paumé, il faut connaître l'informatique, il y en a dans tous les métiers. (Léo, 3<sup>e</sup>)

Ce point de vue ressort aussi très fortement dans les questionnaires, puisqu'à l'affirmation « Apprendre l'informatique permet de trouver plus facilement du travail », 30 % des collégiens ont répondu « Tout à fait d'accord » et 51 % « Plutôt d'accord ».

Les motifs opératoires personnels énoncés par cinq élèves de primaire et quatre collégiens renvoient, ici, au développement de compétences en informatique, perçues comme nécessaires à la réalisation d'activités spécifiques situées dans leur contexte de vie quotidien. On observe, dans les réponses des élèves de primaire, des difficultés à identifier les possibles offerts par la maîtrise de l'informatique; leurs réponses laconiques montrent une difficulté à se projeter en dehors de leurs pratiques enfantines pour lesquelles ils ont développé des compétences d'usage :

Par exemple, quand tu as un travail, pour savoir combien de salaire tu gagnes. Et, aussi, quand tu vas au magasin, s'il n'y a pas d'argent en plus que tu donnes. [...] Tu vas sur un site où il y a une calculatrice. (Maël, CM2)

Une majorité des collégiens affirme qu'« apprendre l'informatique est utile dans la vie quotidienne » (42 % « Tout à fait d'accord » et 28 % « Plutôt d'accord »). Il faut noter cependant, à nouveau, que les compétences informatiques perçues comme nécessaires au quotidien ne semblent pas issues d'un apprentissage scolaire formel.

L'informatique ça peut être juste savoir se servir de son ordinateur, ou aller regarder une vidéo, *streamer*, aller sur les réseaux sociaux bon bah ça oui ça c'est important pour tous les jeunes de mon âge, même pour ceux qui s'intéressent pas à l'informatique pour ceux qui vont pas devenir informaticien. (Théophile, 3<sup>e</sup>)

Enfin, trois élèves et cinq collégiens ont envisagé l'intérêt d'apprendre informatique à l'école à partir de motifs opératoires de type scolaire; il s'agit alors de développer des savoir-faire pour suivre leur scolarité, de se rendre capable de maîtriser certaines tâches dans les classes supérieures :

Ah si pour nos études! Pour plus tard. [...] Si on a des ordinateurs et qu'on ne sait pas le faire aller, ce serait bizarre. [...] Par exemple moi je vais aller à la fac, peut-être pour aller à la fac, on en aura besoin. (Laury, CM1)

Je pense que c'est pas mal pour l'école, je pense que c'est pas mal pour les études, ça permet vraiment de travailler, d'aller chercher les informations pour rendre un travail qui est plus poussé. Ça me servira à ça. L'informatique pour mes études, je pense que si on sait pas y faire avec l'informatique, on peut être très embêté pour tous les travaux qu'on a à rendre au lycée ce sera encore plus important. (Sarrazin, 3<sup>e</sup>)

Concernant maintenant les motifs intrinsèques, ceux-ci ont été évoqués par six élèves de primaire interrogés : apprendre/faire de l'informatique en contexte scolaire procure du plaisir au regard des conditions pratiques de déroulement des enseignements, de leur appétence pour l'outil informatique en lui-même, indépendamment des contenus d'apprentissage :

C'est bien car on a plus tendance à vouloir travailler car des fois quand tu travailles sur une fiche, tu as moins tendance à vouloir travailler tandis que sur l'ordinateur, on a plus envie de travailler. Pour moi, monsieur il nous l'a jamais dit mais ça entraîne les enfants à vouloir plus travailler. (Céline, CM2)

Enfin, relevons que deux élèves de primaire et quatre collégiens n'ont pas identifié de finalités à l'apprentissage de l'informatique à l'école. Ils ont répondu ne pas savoir ou n'avoir rien appris de ce qu'ils maîtrisaient déjà. Dans les questionnaires, 28 % des collégiens déclarant faire de l'informatique au collège, ont indiqué qu'ils n'apprennent « rien », « rien de nouveau », « rien de spécial » ou encore « rien d'intéressant ». Lors des entretiens, ils expliquent que leur culture, leurs pratiques personnelles et/ou l'aide de leur entourage leur apportent plus que l'apprentissage formel dans le cadre scolaire :

Je ne sais pas [...]. Peut-être parce que nos parents, ils nous expliquent déjà comment il faut faire. (Laury, CM2)

Rien, franchement rien d'utile pour moi, rien de nouveau. C'est des trucs que je connais, ou des trucs, c'est pas de l'informatique. (Noé, 5<sup>e</sup>)

Ce point de vue est tout particulièrement partagé par les collégiens qui ont choisi de pratiquer une activité Scratch ou robotique et se projettent dans un métier technique où les compétences en informatique joueront un rôle central.

## Discussion et conclusion

Ce travail se proposait de comprendre les représentations de l'informatique et de son apprentissage que se sont forgés des élèves de la fin de l'école élémentaire et du collège au cours de leur parcours personnel. Il s'agissait également d'identifier la manière dont ces représentations se construisent, évoluent selon leur culture enfantine. Bien que portant sur un nombre limité d'entretiens, les données recueillies ont permis de repérer des ruptures et continuités dans les rapports à l'informatique des élèves en fonction de leur âge, de leur niveau scolaire et des différents contextes d'usage (notamment scolaire et extra-scolaire).

Une première tendance observée est que quels que soient l'âge des élèves, le cycle d'étude et les enseignements suivis, l'informatique est très souvent définie au travers des usages du quotidien. Ces représentations partagées par les élèves du primaire et du secondaire s'ancrent dans une culture numérique influencée par leurs pratiques ordinaires des outils informatisés et ceci, bien qu'il existe deux différences observées entre le primaire et le secondaire. D'une part, on note chez les plus jeunes, un accès aux outils informatisés plus limité, et des pratiques moins diversifiées, l'entrée au collège constituant une période charnière avec l'arrivée du premier smartphone<sup>5</sup> (Crédoc, 2021). D'autre part, en primaire, on observe des pratiques d'enseignement où l'informatique joue la fonction d'outils au service d'autres disciplines et est plus rarement considérée comme objet d'apprentissage tandis qu'au collège cet enseignement commence, si l'on considère les programmes, à être plus technique et scientifique. Nos données de terrain montrent que les sciences informatiques au collège prennent une place plus congrue que prévue dans les programmes, les collégiens déclarant ne pas faire ou faire peu d'informatique en classe. De plus, on observe chez tous les élèves une dissociation entre savoir-faire et savoirs (Wing, 2008), beaucoup considérant que l'informatique renvoie plus à l'action (créer un site Internet, mettre du contenu sur Internet, faire une recherche sur Internet, etc.) qu'à la compréhension des notions ou du fonctionnement (comprendre le fonctionnement d'Internet, par exemple). Ce résultat s'expliquerait par la prédominance des pratiques ordinaires de l'informatique sur les pratiques scolaires. L'école jouerait un rôle mineur dans la (trans)formation des rapports à l'informatique des élèves âgés de 9 à 15 ans. Notons également que cette conception de l'informatique limitée aux usages induit chez les élèves de primaire un sentiment de compétence en informatique : ceux-ci ont développé des compétences d'usage dans leur environnement personnel, par mimesis et/ou tâtonnement, qu'ils jugent suffisantes à leurs pratiques enfantines. Ce sentiment de compétences a tendance à s'effriter lorsqu'ils sont confrontés, au collège, à un enseignement

5. 89 % des collégiens interrogés déclarent posséder leur propre téléphone portable (conformément aux données issues de l'enquête Crédoc (2021) indiquant que 91 % des 12-17 ans étaient équipés de leur propre téléphone portable).

scientifique de l'informatique : leurs représentations s'élargissent, sortant des simples usages, ce qui leur fait prendre conscience, ou, en partie, de la complexité de cet objet.

Une deuxième tendance observée concerne les difficultés éprouvées par la majorité des élèves à adopter une posture de secondarisation (Bautier et Goigoux, 2004), à identifier les enjeux cognitifs de la tâche mettant en jeu l'informatique. Ils ont développé un rapport pratique à l'apprendre. Par exemple, une activité de programmation via Scratch proposée à l'école élémentaire et au collège, donne lieu à des verbalisations portant souvent sur les actions réalisées (« j'ai déplacé le lutin ») et moins sur les savoirs scientifiques mobilisés (« j'ai programmé un déplacement en faisant une répétition et une boucle »). Les élèves sont, ici, focalisés sur la réussite de la tâche sans comprendre ce que cette dernière leur permet d'apprendre (Charlot, 1999). Apprendre l'informatique pour ces élèves, c'est parvenir à réaliser un certain nombre d'actes inhérents à une activité mettant en jeu l'outil informatique. L'absence d'une formation spécifique des enseignants induit des pratiques d'enseignement où l'informatique joue la fonction d'outils au service d'autres disciplines et est plus rarement considérée comme objet d'apprentissage. Et, quand bien même, l'informatique serait travaillée en tant que telle, les difficultés des enseignants eux-mêmes à identifier les savoirs et compétences en jeu dans les séances (Fluckiger *et al.*, 2021), ne leur permettent pas de proposer à leurs élèves des activités variées de manipulations et réflexives, de formaliser ce qui est enseigné. Ancrer les pratiques d'enseignement et les savoirs scolaires dans une démarche théorique (institutionnalisation) mais aussi dans une réflexion plus globale autour de la culture informatique, son utilité à court et moyen termes, ses liens avec les autres disciplines (transferts de compétences) pourrait aider les élèves à rompre avec leur image naïve de l'informatique, bien souvent limitée à l'usage qu'ils en ont. Ce rapport pratique à l'apprentissage est néanmoins porteur de sens chez les élèves, en particulier les plus jeunes d'entre eux ; se rendre capable d'utiliser les outils informatisés répond à leurs besoins quotidiens. Cette valeur attribuée à l'apprentissage de l'informatique pourrait constituer un levier didactique intéressant pour aborder plus explicitement les savoirs technologiques et scientifiques inhérents à ces outils, à leurs pratiques ordinaires.

Une troisième tendance observée porte sur les finalités attribuées à l'apprentissage de l'informatique chez les élèves de l'école de l'élémentaire et du collège ; nos analyses montrent que l'ensemble des élèves mettent en avant des motifs extrinsèques et, plus fortement, de type vocationnel. On note néanmoins, avec l'avancée en âge, une proportion plus marquée de motifs vocationnels chez les élèves. Ce résultat s'explique, chez les plus âgés, par une prise de conscience de la part qu'occupe l'informatique dans des secteurs professionnels variés, à la différence des plus jeunes, qui

ont une vision plus limitée de l'intérêt d'acquérir des compétences en informatique pour leur insertion professionnelle, réduisant, bien souvent cette acquisition au seul métier d'informaticien. Une autre différence observée entre les élèves de l'école élémentaire et du collège concerne le plaisir éprouvé par les premiers au fait d'apprendre/de faire de l'informatique en contexte scolaire. Les élèves les plus jeunes ont développé au cours de leurs expériences scolaires et extra-scolaires un rapport ludique au numérique et à l'apprendre; ce rapport tend à s'atténuer chez les élèves les plus âgés à travers des pratiques scolaires moins récréatives. De plus, nos résultats montrent une dernière rupture entre les deux catégories d'élèves étudiées : un sens attribué à l'apprentissage de l'informatique qui s'effrite avec l'âge. Les plus âgés ont ainsi fait part du peu d'intérêt que présente pour eux l'apprentissage de l'informatique au collège, considérant que leurs pratiques personnelles plus riches sont davantage source de développement de compétences.

Cette contribution souligne l'intérêt de mener des recherches comparatives et/ou longitudinales qui s'intéressent au processus de construction des rapports à l'informatique des élèves, à la manière dont ceux-ci, inscrits dans une culture enfantine en constante évolution, se transforment. Ces recherches ne peuvent se faire que sur un temps résolument long, du primaire jusqu'au secondaire, si l'on souhaite saisir et comprendre les ruptures et continuités dans les rapports à l'informatique des apprenants, la manière dont culture numérique enfantine (ordinaire) et culture scolaire (légitime) se nourrissent, se confrontent et donnent forme aux rapports à l'informatique des élèves.

### **Recommandations**

L'objectif premier de cette recherche était de mieux comprendre les représentations de l'informatique d'élèves de cycles 3 et 4 et la manière dont elles évoluent avec l'âge, en lien avec leur culture enfantine. Il ne s'agissait pas de formuler des préconisations quant à l'enseignement de l'informatique à l'école. Cependant nos résultats soulignent l'intérêt de :

- comprendre les rapports à l'informatique des élèves pour construire une didactique de l'informatique centrée sur le sujet apprenant ;
- identifier des pratiques ordinaires des élèves en fonction de leur âge pour construire des situations d'apprentissage significatives et mobilisantes ancrées dans leur culture enfantine ;
- partir de ces situations pour faire émerger des notions théoriques et scientifiques plus complexes qui leur permettraient de mieux comprendre l'environnement informatisé dans lequel ils évoluent et en avoir un usage conscient et critique.

## Références

- Abric, J.-C. (2003). L'étude expérimentale des représentations sociales, dans D. Jodelet (dir.), *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 203-223. <https://doi.org/10.3917/puf.jodel.2003.01>
- Amadiou, F., et Tricot, A. (2014). *Apprendre avec le numérique : mythes et réalités*, Retz. <https://doi.org/10.14375/NP.9782725633206>
- Arleo, A., et Delalande, J. (2010). *Cultures enfantines : universalité et diversité*, Presses universitaires de Rennes. <https://doi.org/10.4000/books.pur.10732>
- Baron, G.-L., et Boulc'h, L. (2011). Les technologies de l'information et de la communication à l'école primaire. État de question en 2011, *Revue de l'enseignement public et informatique (EPI)*. <https://www.epi.asso.fr/revue/articles/a0910c.htm>
- Baron, G.-L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie*, vol. 195, n° 2, p. 51-62. <https://doi.org/10.4000/rfp.5032>
- Baron, G.-L., Bruillard, E., et Drot-Delange, B. (2015). *Informatique en éducation : perspectives curriculaires et didactiques*, Presses universitaires Blaise-Pascal. <https://hal.science/hal-01136337>
- Bautier, É. (2002). Du rapport au langage : Question d'apprentissages différenciés ou de didactique ? *Pratiques*, vol. 113, n° 1, p. 41-54. <https://doi.org/10.3406/prati.2002.1944>
- Bautier, E., et Goigoux, R. (2004). Difficultés d'apprentissage, processus de secondarisation et pratiques enseignantes : une hypothèse relationnelle, *Revue française de pédagogie*, vol. 148, n° 1, p. 89-100. <https://doi.org/10.3406/rfp.2004.3252>
- Beillerot, J. (1989). « Le rapport au savoir », dans J. Beillerot, A. Bouillet, C. Blanchard-Laville, N. Mosconi et P. Obertelli (dir.), *Savoir et rapport au savoir. Elaboration théorique et clinique*, Éditions universitaires, p. 165-202.
- Bellegarde, K. (2015). *Littéracie et appropriation : ruptures et continuités dans les rapports à l'écrit d'adultes en parcours d'insertion / formation*, thèse de l'université Lille 1. <https://www.theses.fr/2015LIL12012>
- Bellegarde, K., Boyaval, J., et Alvarez, J. (2021). Initier des élèves de maternelle à la robotique/informatique : quand les supports médiateurs impactent la grammaire de l'agir enseignant, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 28, n° 3, p. 13-38. <https://doi.org/10.23709/sticef.28.3.5>

Blaya, C., et Alava, S. (2012). *Risques et sécurité des enfants sur Internet : rapport pour la France-résultats de l'enquête EU Kids Online menée auprès des 9-16 ans et de leurs parents en France.*

<http://eprints.lse.ac.uk/46443/1/FranceReportFrench.pdf>

Bruillard, E. (2009b). Place de l'informatique dans l'enseignement secondaire, *EpiNet : la revue électronique de l'EPI*. <https://edutice.archives-ouvertes.fr/edutice-00558936/file/a0910c.htm>

Bulletin officiel (2015). Socle commun de connaissances, de compétences et de culture, BO n° 17 du 23 avril 2015. <https://www.education.gouv.fr/bo/15/Hebdo17/MENE1506516D.htm>

Carré, P. (1998). Motifs et dynamiques d'engagement en formation. Synthèse d'une étude qualitative de validation auprès de 61 adultes en formation professionnelle continue, *Éducation permanente*, vol. 136, p. 119-131.

Carré, P., et Caspar, P. (2017). *Traité des sciences et des techniques de la Formation, 4<sup>e</sup> éd.*, Dunod. <https://doi.org/10.3917/dunod.carre.2017.01>

Charlot, B. (1997). *Du rapport au savoir. Éléments pour une théorie*, Anthropos.

Charlot, B. (1999). *Le rapport au savoir en milieu populaire*, Anthropos. <https://doi.org/10.4000/books.editions-cnrs.31266>

Charlot, B. (2001). *Les jeunes et le savoir. Perspectives internationales*, Anthropos.

Crédoc (2021). *Baromètre du numérique, édition 2021*. <https://www.credoc.fr/publications/barometre-du-numerique-edition-2021>

Delalande, J. (2003). Culture enfantine et règles de vie, *Terrain. Anthropologie et sciences humaines*, vol. 40, p. 99-114. <https://doi.org/10.4000/terrain.1555>

Durkheim, E. (1987). *Les règles de la méthode sociologique*, PUF.

Fluckiger, C. (2008). L'école à l'épreuve de la culture numérique des élèves, *Revue française de pédagogie. Recherches en éducation*, vol. 163, p. 51-61. <https://doi.org/10.4000/rfp.978>

Fluckiger, C. (2019b). Numérique en formation : des mythes aux approches critiques, *Éducation permanente*, vol. 219, n° 2, p. 19-30. <https://doi.org/10.3917/edpe.219.0019>

Fluckiger, C., et Bart, D. (2012). L'introduction du brevet informatique et Internet (B2i) à l'école primaire : évaluer des compétences hors d'une discipline d'enseignement? *Questions Vives*, vol. 7, n° 17, p. 71-87. <https://doi.org/10.4000/questionsvives.1006>

Fluckiger, C., Haspekian, M., et Grugier, O. (2021). *Enseigner l'informatique à l'école primaire : comment les enseignants voient les sujets didactiques*, Colloque eTIC 4, Caen, France.



Hadopi (2017). *Les 8-14 ans : l'émergence d'une génération de « smartphone natives »*, Hadopi. <https://www.hadopi.fr/actualites/les-8-14-ans-lemergence-dune-generation-de-smartphone-natives>

Ipsos (2017). *Junior Connect' 2017 : les jeunes ont toujours une vie derrière les écrans !* <https://www.ipsos.com/fr-fr/junior-connect-2017-les-jeunes-ont-toujours-une-vie-derriere-les-ecrans>

Jodelet, D. (2003). « Présentation de l'ouvrage », dans *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 5-7. <https://doi.org/10.3917/puf.jodel.2003.01.0005>

Lahire, B. (1993). *La raison des plus faibles : Rapport au travail, écritures domestiques et lectures en milieux populaires*, FeniXX.

Lahire, B. (2008). La forme scolaire dans tous ses états, *Swiss Journal of Educational Research*, vol. 30, n° 2, p. 229-258. <https://doi.org/10.24452/sjer.30.2.4790>

Leclercq, V., Bellegarde, K. (2015). « Appropriation de l'écrit par des adultes faibles lecteurs et faibles scripteurs : rôle des conceptions de la langue et de son apprentissage », dans *Actes du colloque opale, pour un Français convivial : s'approprier la langue*, De Boeck, p. 129-144. <https://doi.org/10.3917/dbu.slfc1.2015.01.0129>

Melin, V. (2019). « Rapport au savoir », dans *Vocabulaire des histoires de vie et de la recherche biographique*, Érès, p. 130-133. <https://doi.org/10.3917/eres.delor.2019.01.0130>

Moscovici, S. (2003). « Des représentations collectives aux représentations sociales : éléments pour une histoire », dans *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 79-103. <https://doi.org/10.3917/puf.jodel.2003.01.0079>

Moskal, B. M., Behrens, N., Guzdial, M., Tew, A. E., Dann, W., et Cooper, S. (2006). Computer science assessment instrument development : Evaluating attitudes and outcomes, *STEM Assessment Conference*, p. 194.

Paillé, P. et Mucchielli, A. (2003). *L'analyse qualitative en sciences humaines et sociales*, Armand Colin.

Reuter, Y. (2007a). « Forme scolaire », dans Y. Reuter (dir.), *Dictionnaire des concepts fondamentaux des didactiques*, De Boeck, p. 11-15.

Conseil scientifique de la SIF (2014). L'informatique : la science au cœur du numérique, *Bulletin de la société informatique de France*, vol. 2, p. 13-20. <https://doi.org/10.48556/SIF.1024.2.13>

Spach, M. (2017). Activités robotiques à l'école primaire et apprentissage de concepts informatiques : quelle place du scénario pédagogique ?

Les limites du co-apprentissage, thèse de l'université Sorbonne Paris Cité. <https://theses.hal.science/tel-02271924>

Vergnaud, G. (1989). « La théorie des champs conceptuels », *Publications de l'Institut de recherche mathématiques de Rennes*, n° S6, p. 47-50.

Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, vol. 366, n° 1881, p. 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>

## Annexe. Représentation de l'informatique des élèves de cycle 3 à travers le concept de classes de situation

Une étude supplémentaire a été menée au sein d'une autre classe du bassin minier. Vingt-trois enfants de CM1/CM2 ont été regroupés en huit groupes, composés d'environ trois élèves. Des *focus groups* ont été réalisés avec ces enfants. Ces *focus groups* avaient pour objectif de les faire verbaliser sur les outils numériques possédés mais aussi sur les connaissances et les pratiques associées à ces outils.

La première partie des *focus groups* questionnait les enfants sur leur équipement en outils numériques et sur leurs modalités d'utilisation de ces appareils. Au fur et à mesure des réponses des enfants, des cartes mentionnant des tâches numériques citées par les enfants étaient créées. Lors de la seconde partie, les enfants devaient organiser ces cartes en familles (soit en classes de situations au sens de Vergnaud), et justifier leur classement.

Au total, les enfants ont formé 49 classes de situation, regroupant une à six cartes, soit une à six tâches et en ont justifié 44. L'analyse de ces 44 justifications montre que les représentations des enfants portent avant tout sur les usages du numérique, en témoignent l'écrasante majorité des justifications des enfants associées à des usages du numérique ( $n = 38$ , soit 86 %). Ce n'est que très partiellement que les enfants font appel à leurs connaissances portant sur la dimension scientifique ou technique de l'informatique (pour chacune des dimensions,  $n = 3$ , soit 7 %).

### Associer des tâches selon ses usages du numérique

Selon cette logique, les enfants justifient l'association de tâches en une même classe de situation par l'identification d'usages communs entre les tâches mentionnées sur les cartes. C'est par exemple le cas de Louna qui associe les tâches *écouter de la musique, jouer sur la Switch, jouer sur le smartphone, jouer sur la PS4, installer une nouvelle application et regarder une vidéo YouTube* :

Jouer sur la switch on peut mettre de la musique, jouer sur la PS4 on peut en mettre, et jouer sur un smartphone on peut en mettre et regarder une vidéo, bin on peut regarder sur tout et installer une nouvelle application on peut installer sur tout. (Louna)

Louna associe les différentes tâches proposées selon ses propres usages des outils numériques. Elle écoute de la musique aussi bien sur sa Switch que sur la PS4 ou encore le smartphone. Dans le

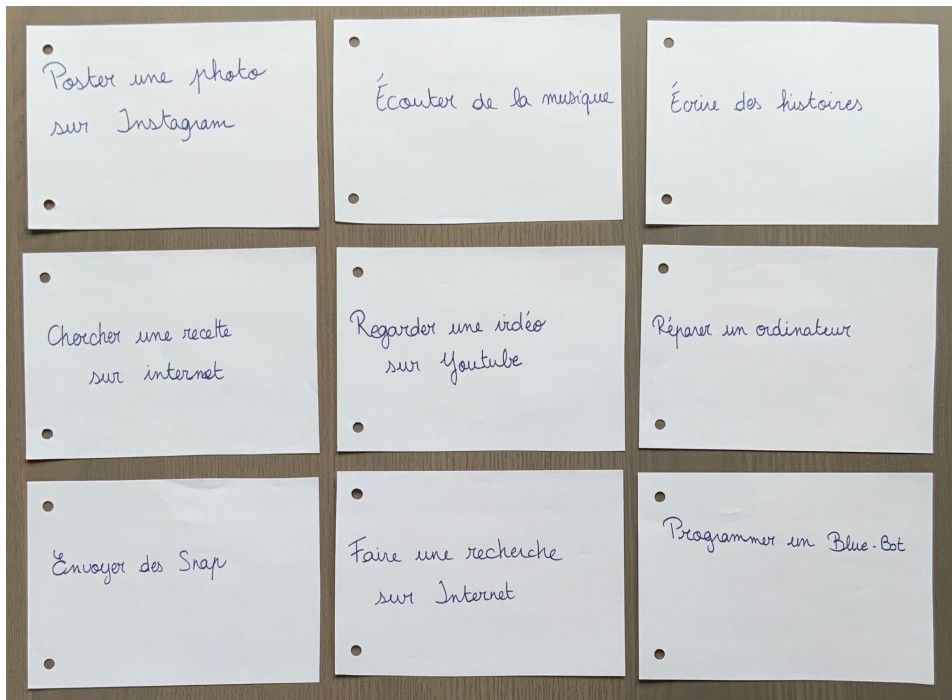


FIGURE 1 – Aperçu de quelques cartes utilisées lors des *focus groups*.

même ordre d'idées, Garance associe les tâches *jouer sur la Switch*, *jouer sur la PS4*, *regarder une vidéo YouTube* et *regarder Netflix* en mentionnant qu'elle consulte YouTube et Netflix via la PS4 et via la Switch. Les usages des outils numériques mentionnés par les enfants sont de natures diverses (jouer, communiquer, écrire, etc.).

### Associer des tâches selon ses connaissances en science informatique

Selon cette logique, les enfants justifient l'association de tâches en une même classe de situation en mettant en avant leurs connaissances en science informatique. Ainsi, Charles associe les tâches *faire une mise à jour de logiciel*, *installer une nouvelle application* et *programmer un Bluebot* en mobilisant le concept d'exécution :

Déjà programmer le Bluebot c'est par exemple t'as [en montrant sur la table les différentes zones où se situent les flèches] une flèche une autre flèche une autre flèche à droite à gauche devant derrière et tu dois programmer donc euh comme installer un nouveau logiciel tu dois taper ton logiciel le programmer et pouvoir après le lancer. (Charles)

Dans cet exemple, Charles met en avant l'idée que ses actions modifient quelque chose sur la machine, qu'il est acteur et dialogue

avec la machine. Cela est à associer au concept informatique d'exécution : il faut « pouvoir après le lancer ».

### Associer des tâches selon ses connaissances technologiques

Selon cette logique, les enfants justifient l'association de tâches en une même classe de situation en se référant à la dimension technologique de l'informatique : ils mobilisent notamment leurs connaissances portant sur le fonctionnement des machines, sur l'architecture des dispositifs informatiques, etc. C'est par exemple le cas d'Ethan qui différencie la classe de situations constituée des tâches *jouer à la PS4* et *jouer à la Switch* de la classe constituée de *prendre une photo* :

Prendre une photo t'as juste à appuyer sur un bouton et ça le fait alors que si tu joues sur la PS4 ou la Switch y a des joysticks c'est tout un système. (Ethan)

Ethan effectue son classement en distinguant des appareils numériques mobilisés, il distingue ainsi les consoles de jeu et les appareils photos. Alors qu'il perçoit les consoles comme des objets complexes, « c'est tout un système », l'appareil photo est perçu par Ethan comme un objet plus simple où il faut « juste appuyer sur un bouton ».

Dans le même ordre d'idées, Charles mobilise aussi des connaissances se référant à la technologie lorsqu'il justifie sa classe de situation formée de la tâche *réparer un ordinateur* :

Tu dois changer les câbles qu'il y a dedans [dans l'ordinateur] voir s'ils sont détachés coupés tu dois racheter des câbles et après remettre tout bien [. . .] y a le cerveau de l'ordinateur qui commande tous les fils. (Charles)

Au travers de sa justification, Charles met en avant ses connaissances portant sur les composants internes de l'ordinateur : « le cerveau de l'ordinateur commande les fils ».

Pour citer ce chapitre :

Bellegarde, Katell, Boulc'h, Laetitia, et Vandeveld, Isabelle (2024). « Ruptures et continuités dans les représentations de l'informatique et de son apprentissage chez les élèves de cycles 3 et 4 », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 87-114. <https://doi.org/10.53480/2024iecare05s/>

# Usages des robots programmables BeeBot en classe : des objets pour apprendre, des objets à apprendre

# 6

Sandra NOGRY<sup>1</sup>  
Olivier GRUGIER<sup>2,3</sup>

1. Cergy Paris Université, INSPE de Versailles, Laboratoire EMA, 95000 Cergy-Pontoise, France

2. Sorbonne Université, INSPE de l'Académie de Paris, 75017 Paris, France

3. Université Paris Cité, Laboratoire EDA, 75006 Paris, France

Pour enseigner de la programmation auprès de jeunes élèves dans les écoles en France, certains enseignants introduisent des robots dans leur classe.

Ce chapitre propose d'analyser les obstacles à l'appropriation de ces robots rencontrés par les élèves de ces classes avant de mettre en évidence les médiations qui soutiennent ce processus. Nous montrerons que l'introduction de robots programmables renvoie à deux objectifs : apprendre à les utiliser pour apprendre à programmer.

## Robots en classe pour apprendre la programmation

À la suite des premiers travaux de Papert (1980) portant sur la programmation avec la tortue Logo, puis suite à la réintroduction de l'informatique dans les curricula de l'école primaire (Baron et Drot-Delange, 2016), des robots planchers programmables sont introduits pour enseigner la programmation dans les écoles (Greff, 1999). L'intérêt de tels objets manipulables et programmables pour faire découvrir aux jeunes élèves les bases de l'informatique a été démontré (Benitti, 2012 ; Toh *et al.*, 2016). Depuis une dizaine d'années, plusieurs travaux de recherche ont montré que la robotique pédagogique est possible dès les premières années de la scolarisation pour apprendre différents concepts et habiletés centrales en informatique tels que la capacité à construire une séquence d'instructions en sélectionnant et en ordonnant un ensemble d'instructions pour atteindre l'objectif visé. En apprenant à programmer de tels objets, les apprenants développent leur capacité de réflexion, d'observation et construisent des stratégies de résolution de problèmes (Bers *et al.*, 2014 ; Kazakoff *et al.*, 2013 ; Komis et Misirli, 2015 ; Komis *et al.*, 2017).

Dans une perspective constructiviste, il s'agit de fonder la construction de savoirs sur l'expérience directe de l'individu médiée par la manipulation de cet objet tangible ; programmer un robot de sol exige que l'on réfléchisse d'abord à la façon dont on ferait soi-même ce qu'on désire lui faire faire avant de transposer ces actions dans un programme. La programmation de ces objets tangibles est ainsi conçue à la fois comme une abstraction dans un langage de programmation, et comme la concrétisation de notions informatiques par les déplacements effectifs du robot : « Les procédures acquièrent une existence reconnue, on les nomme, on les manie, on les identifie, et c'est ainsi que les enfants, progressivement, en viennent à se forger la notion de procédure » (Papert, 1980, p. 192).

Dès l'apparition des premiers robots programmables destinés à un apprentissage dans un cadre scolaire, ceux-ci ont été conçus selon une approche anthropomorphique. Comme le soulignait Papert (1980), un des premiers robots introduit dans des classes, la tortue, a une position et une orientation, tout comme un être humain : « les enfants peuvent s'identifier à la tortue, et faire ainsi appel à la connaissance qu'ils ont de leur corps et de son mouvement pour aborder la géométrie formelle » (Papert, 1980, p. 76). Les concepteurs en ont fait des objets ludiques et attractifs avec la présence d'yeux circulaires, d'un large sourire et disposant d'un nom faisant référence à des animaux comme la tortue dans les années 80 et qui a laissé sa place aux « abeilles » depuis la réintroduction d'un enseignement de l'informatique dans les programmes de l'école en France. Ces objets semblent néanmoins faire apparaître des confusions mentales chez les plus jeunes élèves (Grugier, 2022). Un objet conçu pour se déplacer au sol avec des roues et qui porte le nom d'« abeille » renvoie à deux références sociales sans lien évident chez les élèves. En effet, il s'agit d'un objet roulant comme une voiture mais qui est appelé une abeille comme un insecte qui vole ! Le caractère anthropomorphique est également présent dans les mouvements des robots (Duboulay, 1986, cité par Rogalski, 2015 ; Levy et Mioduser, 2008) : la machine paraît prendre en compte les intentions, traiter sémantiquement les opérations en jeu ou partager certaines capacités humaines considérées comme allant de soi ; la compréhension du fonctionnement effectif de ces objets programmables nécessite un apprentissage.

Ces objets ont ainsi un double statut (Lebeaume, 2019 ; Bisault *et al.*, 2023) : objet tangible pour apprendre de nouvelles procédures ou différentes notions, et objet technique à apprendre pour agir rationnellement avec ce dernier. Dans ce chapitre, nous interrogeons les modalités d'appropriation de ces objets programmables par les élèves à partir de l'analyse de l'activité de ces apprenants en classe lors de pratiques ordinaires d'enseignement-apprentissage. Les travaux pionniers ont montré que de jeunes élèves étaient capables d'apprendre à programmer dans des classes avec des

enseignants spécialement formés. Cette contribution souhaite explorer l'appropriation des robots par des élèves dans des classes ordinaires et apporter des réponses à plusieurs questions : comment les élèves s'approprient-ils / apprennent-ils à utiliser les objets programmables à travers les situations d'enseignement-apprentissage proposées et en fonction de leur âge ? Quels sont les obstacles à cette appropriation / à cet apprentissage ? Quel(le)s sont les médiations / dispositifs d'enseignement-apprentissage qui soutiennent cette appropriation ?

À partir d'une synthèse de plusieurs études de cas dans lesquelles nous avons analysé l'activité instrumentée des élèves dans des classes de cycle 1, à cycle 3 de l'école primaire (Grugier, 2021; Grugier et Nogry, 2022; Nogry et Spach, 2022), nous montrerons qu'en fonction de leur âge et de leur niveau scolaire, les obstacles rencontrés et les démarches d'appropriation diffèrent. Dans un second temps, nous analyserons différentes médiations mises en œuvre par les enseignants pour soutenir cette appropriation.

## **Robots planchers fréquemment utilisés dans le cadre scolaire**

Les fournisseurs de matériel pédagogique proposent différents robots plancher qui utilisent, comme les véhicules de Braitenberg (1984) et la tortue de chez Jeulin, deux roues motrices permettant de générer le déplacement. Parmi les onze principaux fournisseurs de matériels pédagogiques pour les écoles françaises, quatre ont accepté de nous communiquer les robots plancher qui sont les plus vendus. Les deux robots les plus demandés par les écoles et les collectivités territoriales en charge des équipements scolaires sont les robots BeeBot et BlueBot, avec des ventes qui représentent entre 60 % et 90 % du marché des robots. Le robot Ozobot puis le robot Thymio arrivent loin derrière avec au maximum 15 % de parts de marché.

En partant de ce constat, nous sommes allés observer des pratiques scolaires avec les robots BeeBot et BlueBot. Ces deux robots sont identiques en termes de déplacement et de programmation. Ils se programment à partir d'une interface tangible disposée sur leur partie supérieure (figure 1). Quatre touches de couleur orange et une de couleur bleue permettent d'incrémenter des instructions pour le déplacement (« Avancer », « Reculer », « Pivoter à droite », « Pivoter à gauche », « Pause » : interruption momentanée de l'exécution du programme), deux touches permettent de contrôler l'exécution du programme (« Go » : exécution des instructions programmées, « Clear » : effacement des commandes stockées en mémoire).



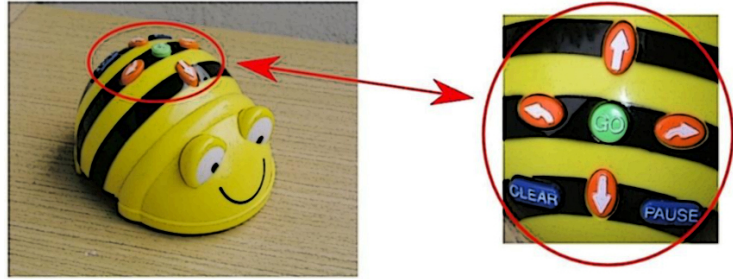


FIGURE 1 – Interface tangible d'un robot BeeBot.

## Robot Beebot : un objet à apprendre

Afin d'étudier l'appropriation de ces objets programmables, nous avons analysé huit séquences d'apprentissages observées dans différents contextes (tableau 1), en mobilisant l'approche instrumentale (Rabardel, 1995). Cette approche anthropocentrée qui désigne les objets techniques (et par extension tout objet ayant subi une transformation) sous le terme générique d'*artefact*, offre un cadre adapté pour étudier comment l'artefact devient instrument :

L'instrument comprend dans cette perspective un artefact matériel ou symbolique produit par l'utilisateur ou par d'autres ; un ou des schèmes d'utilisation associés résultant d'une construction propre ou de l'appropriation de schèmes sociaux préexistants [ . . . ]. Ensemble, artefact et schèmes constituent l'instrument. (Rabardel, 1995)

Cette analyse vise ainsi à mettre en évidence les genèses instrumentales en documentant les éventuelles modifications apportées à l'artefact (instrumentalisation) et la constitution de schèmes associés à l'artefact (instrumentation). Les schèmes – organisations invariantes de l'action pour une classe de situation – rendent compte des relations entre le geste et la pensée développées par les élèves (Rabardel, 1995). Nous avons porté une attention particulière aux schèmes d'utilisation progressivement associés au robot pour faire de ces artefacts des instruments, qu'il s'agisse des schèmes d'usage, correspondant à l'interaction du sujet avec l'artefact, ou des schèmes d'action instrumentés qui s'appuient sur les schèmes d'usages pour atteindre les buts poursuivis.

Les séquences observées et filmées avaient pour objectif de faire découvrir le fonctionnement du robot BeeBot, de construire un algorithme et d'apprendre à programmer l'objet afin de générer un déplacement souhaité. La complexité des parcours nécessitait un déplacement rectiligne sur un plateau matérialisé par des cases, ou un déplacement avec des changements de direction pour les élèves les plus grands. Sur le plan pédagogique, des démarches

TABLEAU 1 – Répartition et caractéristiques des séquences d'apprentissage observées.

Cycle	Niveau de classe	Période d'observation	Nombre de séances et durée	Taille des groupes constitués
1	PS	Octobre et novembre 2017	5 de 30 à 40 min	2 à 4 élèves
1	PS	Octobre et novembre 2018	5 de 30 à 40 min	2 à 4 élèves
1	GS	Octobre et novembre 2020	5 de 30 à 40 min	2 à 3 élèves
1	GS	Octobre et novembre 2021	5 de 30 à 40 min	2 à 3 élèves
1	GS	Octobre et novembre 2022	5 de 30 à 40 min	2 à 3 élèves
2	CP	Janvier à mars 2017	6 de 30 à 50 mn	3 élèves
2	CE1	Janvier à mai 2017	15 de 30 à 50 mn	2 à 4 élèves
3	CMI	En février et avril 2018	5 de 50 à 60 min	4 élèves

d'investigation favorisant la manipulation et l'expérimentation collective, ont été privilégiées. Par groupes, les élèves sont mis en présence de BeeBot qu'ils découvrent, manipulent et programment pour générer des déplacements.

Les observations ont été menées dans des classes ordinaires auprès d'enseignants non spécialistes et non formés à l'utilisation du robot mais volontaires pour développer des apprentissages de programmation avec le robot programmable BeeBot auprès de leurs élèves. Ces classes (tableau 1), en région parisienne, comprennent entre 24 et 28 élèves en fonction des niveaux, ce qui représente environ 200 élèves. Les groupes d'élèves ont été librement constitués par les enseignants sans consigne du chercheur. Les séances ont été filmées à partir de matériel nomade pour réduire l'impact de l'intrusion sur le déroulement des séances. Le corpus est constitué de plus de 33 h de vidéo. Dans ce corpus, ce sont les moments scolaires d'appropriation et de construction mentale qui ont été principalement analysés. Aucune distinction n'a été faite, dans l'analyse, entre les filles et les garçons. Enfin, les élèves observés découvraient tous pour la première année le robot BeeBot. Il n'a pas été mené d'observation sur plusieurs années d'une cohorte, ce qui aurait peut-être permis d'identifier les modifications dans la manière de penser ces objets.

### **Se familiariser avec l'objet technique par la manipulation en groupe**

La première phase de ces séquences est principalement centrée sur la découverte du fonctionnement du robot en adoptant une démarche d'investigation. La durée de cette phase varie fortement, entre 15 minutes en cycle 3 et plusieurs séances de 40 minutes dans les classes de petite section de maternelle.

Dans les classes du cycle 1, la première séance est focalisée sur la rencontre et la découverte du robot. Les élèves observés découvrent

pour la première fois cet objet qui est nouveau pour eux. C'est la manipulation directe qui est privilégiée. Cette familiarisation conduit les jeunes élèves à s'apercevoir que les robots possèdent des roues. Ils déplacent alors le robot sur la table en le poussant, en référence à des jouets fonctionnant sur ce principe, mais les roues frottent sur le support. Après quelques secondes, les élèves se mettent à appuyer sur les touches de l'interface de programmation. L'objet ne se déplace toujours pas et ne laisse entrevoir aucun fonctionnement. Il faut attendre une nouvelle phase de découverte pour que ces élèves identifient les deux interrupteurs placés sous le robot. Une manipulation de ces deux organes de commande, va mettre en action des voyants lumineux et le bipteur du robot. Toujours dans cette même première séance, les apprenants vont ensuite appuyer sur les touches de l'interface qui vont permettre de générer les premiers déplacements. Les élèves découvrent ainsi que le robot peut avancer sans qu'il soit nécessaire de le pousser. Ceci devient, pour la suite des différentes séances, une opinion socialement partagée par les élèves. Pendant la suite de la première séance, les élèves agissent sur les différentes touches de l'interface mais le robot ne se déplace pas comme ce qui est demandé par l'enseignant. Cependant, les élèves agissent systématiquement sur les deux interrupteurs avant de commencer à l'utiliser. Cette manipulation sur les interrupteurs devient un invariant opératoire.

Au cycle 3, lors de la première séance, l'enseignante dispose au milieu de chaque groupe d'élèves, un robot BeeBot dans sa boîte d'emballage. Dans les groupes, plusieurs organisations se mettent en place. Certains élèves se trouvent lecteur de la notice d'utilisation, d'autres manipulent directement le robot et d'autres occupent une posture d'observateur. Très rapidement, la manipulation directe conduit à découvrir qu'il faut dans un premier temps agir sur les interrupteurs pour mettre en fonctionnement le robot puis, dans un second temps, appuyer sur les touches de l'interface pour générer des déplacements du robot. Contrairement aux élèves plus jeunes, les premières représentations des élèves sur le fonctionnement du BeeBot sont associées à un objet devant être programmé. Les actions sur les interrupteurs puis sur les touches engendrent un déplacement de l'objet. Les toutes premières manipulations permettent aux élèves de se construire un schème d'action instrumenté basé sur un algorithme : agir sur les interrupteurs, agir sur les touches de programmations orange puis agir sur la touche « Go » d'exécution du programme. Néanmoins, lors de cette première séance, il n'y a pas encore de schème stabilisé permettant d'assurer les déplacements souhaités du robot.

Au cycle 2, dans les séances observées, il n'y a pas de séance dédiée spécifiquement à la découverte du robot, l'enseignant montre comment mettre le robot en marche et invite les élèves à

lui faire réaliser un déplacement. La tâche des élèves, réunis par groupes, consiste donc à programmer directement le robot pour qu'il parcoure un chemin préalablement défini par eux-mêmes ; ce qui les conduit à porter leur attention directement sur l'interface de commande.

### Apprendre à contrôler les déplacements du robot : vers la constitution de premiers schèmes d'utilisation

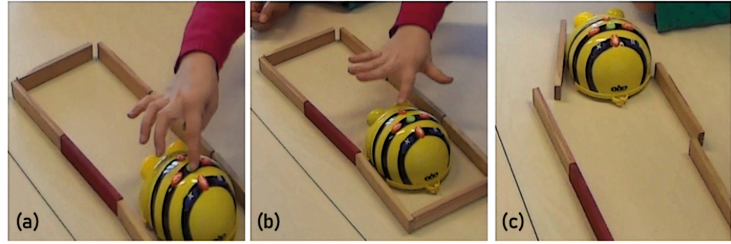
Une fois le fonctionnement du robot découvert, l'activité des élèves se focalise sur la programmation du robot pour que ce dernier se déplace d'un point de départ vers un point d'arrivée en mettant en œuvre une démarche d'investigation. Au cycle 1, au cours des différents essais réalisés par les élèves pour contrôler le déplacement du BeeBot en utilisant l'interface de commande, deux schèmes se stabilisent progressivement (Grugier et Nogry, 2022 ; Grugier, 2022). L'un consiste à appuyer sur les touches de direction puis sur « Go », l'autre consiste à appuyer successivement sur les touches « Go », « Flèche », « Go ». Ces deux schèmes d'utilisation ne permettent toutefois pas aux élèves de contrôler le déplacement du robot dans la direction souhaitée (tableau 2). La découverte du fonctionnement du robot au sein d'une organisation collective conduit à la construction de plusieurs schèmes dans un même groupe.

En cycle 2, les élèves cherchent comment contrôler les déplacements du robot pour générer un mouvement dans la direction et le sens souhaités. L'analyse de l'activité collective des élèves met en évidence l'apparition rapide d'une première genèse instrumentale, un premier schème de programmation pas à pas (Nogry et Spach, 2022) après la phase de découverte du robot. Il s'agit d'appuyer

TABLEAU 2 – Construction de plusieurs schèmes dans une découverte collective en cycle 1.

Elève	Séquence d'action	Déplacement attendu	Observation du robot
1	Manipuler les interrupteurs, Appuyer sur Go, Appuyer sur une instruction de déplacement	Déplacement du robot	Aucun déplacement, le robot émet un son et les voyants s'allument
2	Appuyer sur Go	Déplacement du robot	Le robot se déplace de un pas, en fonction de l'instruction programmée par l'élève 1
3	Appuyer sur une instruction de déplacement puis appuyer sur Go	Déplacement du robot	Le robot se déplace de deux pas, en fonction des deux instructions programmées (élève 1 et élève 3)

**FIGURE 2** – Illustration du schème « Go-Flèche » : séquence d'action fréquemment réalisée pour programmer le déplacement du robot. (a) L'élève appuie sur Go. (b) L'élève appuie sur Flèche. (c) Le robot parcourt un trajet plus long que prévu suscitant surprise et émotion.



sur « Go » puis sur une flèche de déplacement en fonction du sens souhaité. Nous nommerons ce schème « Go-Flèche » (figure 2).

En CP, ce schème – correspondant à une tentative de contrôle direct du robot – est partagé collectivement au sein de la classe lors d'un regroupement, et rapidement mis en œuvre par tous les élèves. Ce faisant, les élèves semblent adopter une conception anthropomorphique du robot selon laquelle la machine traite sémantiquement les opérations en jeu (Rogalski, 2015), ce schème pouvant être interprété comme « va en avant / en arrière / etc. ». Néanmoins, ils constatent rapidement qu'il ne permet pas de contrôler efficacement ses déplacements.

Dans les classes de cycle 1 et de CP observées, les obstacles rencontrés durant les premières séances sont multiples, les élèves ont des difficultés à interpréter le comportement du robot ; nous constatons également un manque de compréhension des fonctions des commandes de contrôle, et notamment des touches « Go » et « Clear ». En effet, la commande « Go » exécute une séquence d'instructions précédemment implémentée et stockée en mémoire. Les instructions stockées dans la mémoire seront effacées si la touche « Clear » est actionnée ou si le robot est éteint par une action sur les interrupteurs. Sinon, toutes nouvelles instructions s'ajoutent aux précédentes dans la mémoire et une action sur « Go » entraîne l'exécution de l'ensemble.

Ainsi, pour les élèves de cycle 1 et 2, le caractère différé de l'exécution du programme – le passage du « faire » au « faire exécuter » une séquence d'instructions – fondé sur le stockage en mémoire des instructions pose des difficultés, et conduit à la construction d'un schème d'utilisation inefficace.

Suite aux premières difficultés et au constat d'inefficacité des premiers schèmes mobilisés, les questionnements des enseignants auprès de chaque groupe ou lors des moments de regroupement, amènent progressivement les élèves à déconstruire ces schèmes en s'interrogeant sur les relations de cause à effet entre les touches sélectionnées et le fonctionnement des actionneurs. Ils attribuent ainsi une fonction aux différentes commandes de contrôle. D'autres questions permettent d'introduire le caractère reproductible du programme. Dans certaines classes de cycle 1 et de cycle 2, ce sont les enseignants qui finissent par indiquer la séquence

de commandes à mettre en œuvre pour effacer le programme précédent, programmer une nouvelle séquence d'instructions puis lancer un nouveau programme. Ainsi, les enseignants proposent la séquence suivante : effacer, programmer des instructions de déplacement puis lancer l'exécution du programme.

Au cycle 3, les fonctions des commandes de contrôle sont rapidement découvertes par les élèves durant les deux premières séances, lors de leurs investigations. Des moments d'institutionnalisation permettent de partager les savoirs découverts. Deux caractéristiques du robot sont rapidement identifiées : celle concernant le stockage des instructions avec la fonction d'effacer en cas d'erreur de programmation ou de modification et celle de longueur de pas de déplacement du robot. En utilisant une règle introduite par l'enseignante, les élèves ont découvert que la longueur d'un pas du robot est d'environ 15 cm. Les questionnements de cette enseignante permettent ainsi aux élèves de construire collectivement un schème commun de fonctionnement.

### **Vers la programmation d'une séquence d'instructions par l'introduction d'artefacts pédagogiques**

Une fois que les élèves sont en capacité d'utiliser les commandes de contrôle du robot, guidés par l'enseignant (maternelle, CP) ou en autonomie (CE1, CM1), un nouvel enjeu concerne la programmation d'une séquence d'instruction.

En cycles 1 et 2, il s'agit pour les élèves d'abandonner progressivement la stratégie « pas à pas », stratégie privilégiée consistant à saisir chaque déplacement et à l'exécuter immédiatement (Komis et Misrili, 2015) pour saisir l'ensemble des pas du trajet avant de demander l'exécution du programme. Cette stratégie nécessite d'anticiper la séquence à produire et de différer son exécution. Dans ce but, les enseignants de cycle 1 et 2 introduisent différentes médiations.

En cycle 1, les enseignants introduisent des plateaux de déplacement sur lesquels des carrés de 15 cm par 15 cm sont dessinés. Ces carrés matérialisent les longueurs de pas de déplacement du robot, même si cette longueur de déplacement n'est pas explicitée. C'est l'approche numérative qui est utilisée (figure 3). Par la suite, afin de représenter les instructions à programmer, ils introduisent un ensemble de « flèches » analogues aux commandes de contrôle du BeeBot (figure 3). Les élèves doivent reproduire sur le BeeBot la séquence d'instructions proposée par l'enseignant, et représentée par une succession de symboles. De fait les élèves rencontrent des difficultés à s'approprier ces symboles. Cette activité est médiée par l'enseignant, qui, par des pointages et des verbalisations, met

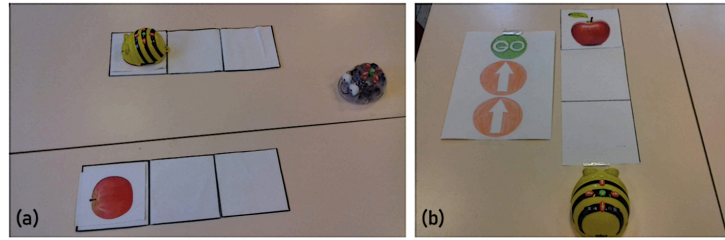
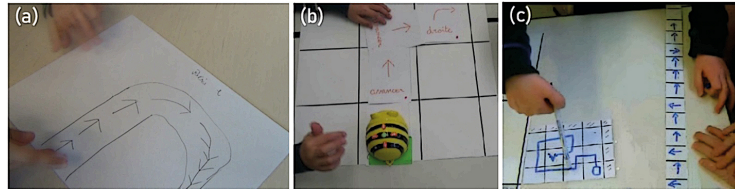


FIGURE 3 – Matérialisation des pas de déplacements en cycle 1.

FIGURE 4 – Artefacts utilisés par les élèves pour programmer le robot (Nogry et Spach, 2022). (a) Schéma construit collectivement par les élèves. (b) Cartes figurant les commandes. (c) « Bande algorithmique » et « petit plateau ».



explicitement en correspondance les représentations symboliques avec les commandes de contrôle du robot. Ces médiations seront progressivement prises en charge par certains élèves au sein des groupes pour aller vers la programmation de parcours plus élaborés. Une stratégie pas à pas perdure néanmoins chez certains élèves.

En CP et CE1, pour passer d'une stratégie pas à pas à la programmation d'une séquence d'instruction, les enseignants introduisent dans le scénario pédagogique une phase de planification des déplacements du BeeBot. Celle-ci vise à différer l'exécution de la séquence d'instruction. Il s'agit pour les élèves de produire des représentations sémiotiques des déplacements et/ou de la séquence d'instruction en mobilisant différents artefacts mis à leur disposition (figure 4a,b), ou en se référant à des règles ou normes collectivement constituées (figure 4a).

Ainsi, dans la classe de CP, les élèves ont à produire par groupe une représentation schématique du trajet à parcourir (figure 4). En réfléchissant collectivement aux conventions à adopter pour que leur schéma soit compris par d'autres groupes, ils se mettent progressivement d'accord sur la symbolisation d'un pas ou d'une rotation. Ces schémas deviennent ensuite un instrument structurant pour programmer.

Dans la classe de CE1 observée, un grand plateau est d'abord proposé, rapidement utilisé par les élèves pour simuler le déplacement du robot en le déplaçant manuellement (figure 4). Devant la difficulté des élèves à se représenter la séquence à programmer, un petit plateau est ensuite introduit afin de schématiser ce parcours en deux dimensions. Cet artefact facilement manipulable facilite la co-construction du parcours au sein du groupe. Néanmoins, cette représentation en deux dimensions n'est pas suffisante pour anticiper la direction des rotations à effectuer, ou pour identifier le nombre de pas à programmer. Un troisième artefact est introduit,



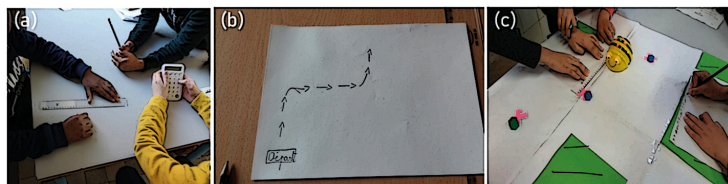


FIGURE 5 – Artefacts utilisés en mathématiques pour programmer le robot. (a) Artefacts usuels utilisés en mathématiques. (b) Schéma construit par les élèves. (c) Plateau et pions.

« la bande algorithmique » (figure 4). De nombreux échanges entre élèves au sein des groupes participent à leur appropriation. Ces artefacts sont progressivement utilisés pour représenter le trajet à parcourir et/ou la séquence d'instructions puis comme des instruments pour programmer des parcours complexes (composés d'un grand nombre de pas et de multiples rotations) ou pour debugger. Une fois la programmation d'une séquence complexe maîtrisée, l'enseignante introduit une nouvelle tâche : programmer les déplacements simultanés de deux robots sur un même quadrillage. En adaptant leurs instruments pour réaliser cette tâche, ils découvrent les enjeux de la programmation parallèle (voir Spach, 2017 pour une description plus complète).

Dans la classe de CM1, l'enseignante introduit d'autres artefacts usuellement utilisés dans l'enseignement des mathématiques (règle, calculatrice) afin de définir le parcours de déplacement que doit faire le robot sur le plateau constitué de feuilles, de pions, d'un point de départ, d'un point d'arrivée et des obstacles à contourner (figure 5).

En cycles 2 et 3, ces artefacts matériels et symboliques deviennent progressivement des instruments pour programmer le robot, observer l'écart éventuel entre le parcours prévu et réalisé, et corriger leur programme.

## De multiples obstacles à la programmation d'une séquence d'instructions

Comme nous l'avons mis en évidence précédemment, en cycle 1 et 2, de nombreux obstacles sont rencontrés par les élèves au cours de leur investigation. Le premier concerne la difficulté des élèves à se représenter le caractère différé de l'exécution du programme préalablement stocké en mémoire. Cet obstacle de type ontogénique est sous-tendu par les spécificités du développement cognitif ; les fonctions exécutives telles que la planification (élaboration d'un plan d'action en fonction d'un but à atteindre) se développent de façon prolongée tout au long de l'enfance et de l'adolescence (Roy, 2015). Il est par ailleurs à noter que l'absence d'interface permettant de visualiser les instructions en mémoire ne facilite pas cette compréhension. Pour dépasser cette difficulté, les enseignants proposent de différer l'exécution du programme



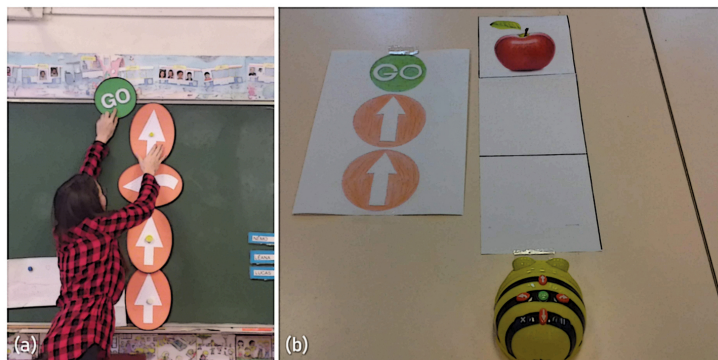
par l'introduction d'une étape de planification médiée par différents artefacts qui visent à représenter le parcours de déplacement et/ou la séquence d'instructions.

D'autres obstacles portent plus spécifiquement sur le passage de la représentation du parcours à effectuer par le robot à la programmation de la séquence d'instructions.

Dans les classes observées en cycle 1 et cycle 2, un obstacle porte sur la programmation du sens des rotations. Celui-ci vient en premier lieu de la difficulté qu'ont les enfants à passer d'une perception de l'espace de déplacement allocentrée (représentation de la trajectoire du robot dans un espace en deux dimensions) à une représentation des déplacements égocentrée (adoption du point de vue du robot). D'après Papert (1980), cette construction de différentes représentations de l'espace passe par une connaissance des mouvements de leur corps lorsqu'ils incarnent les mouvements du robot. Ainsi, nous observons que les enfants utilisent de manière répétée leur corps pour simuler différentes possibilités d'action. De cette façon, ils incorporent différentes stratégies leur permettant de conceptualiser ses déplacements. Dans deux classes de cycle 1, cette incarnation des mouvements du robot est didactisée : l'enseignant propose des séances de motricité pour travailler la notion de déplacement et de pivotement pour effectuer des liens avec les mouvements du robot.

Par ailleurs, une difficulté est causée par l'iconographie choisie pour représenter les rotations sur l'interface de commande du BeeBot (figure 1), source d'incompréhension relative à la nature des déplacements. Elle laisse supposer que cette touche active simultanément une rotation et un déplacement d'un pas. Or, la commande engendre non pas un déplacement mais un pivotement de 90° du robot. Cette non-congruence entre représentation et action provoque des erreurs fréquentes de comptage du nombre de pas à parcourir et de choix du sens de la rotation.

Un autre obstacle à la programmation réside dans une certaine confusion ou un manque de distinction entre parcours de déplacement et séquence d'instructions. Ainsi, en cycle 1 – où l'enseignant utilise un ensemble d'artefacts symbolisant les commandes de contrôles ordonnés suivant les trajets à faire parcourir au robot – la confusion se manifeste tant par le vocabulaire utilisé pour dénommer cette succession – parfois appelée algorithme, parfois « le chemin à suivre » ou encore « le programme » – que par leur orientation en fonction des modalités de présentation. Pendant les phases de regroupement, lorsque l'enseignant s'adresse à la classe entière, ces symboles sont présentés au tableau du bas vers le haut, alors que lorsque les élèves travaillent par groupe sur les tables, ceux-ci sont présentés à l'horizontale, juxtaposés au trajet à parcourir (figure 6), par analogie avec le parcours à effectuer. Or, pour représenter une séquence d'instructions, on pourrait s'attendre



**FIGURE 6** – Représentations symboliques de parcours de déplacements en cycle 1. (a) Présentation des symboles utilisés pour représenter le parcours du robot. (b) Représentation d'un parcours à programmer.

à ce qu'ils soient présentés de gauche à droite, dans le sens de la lecture quelle que soit la situation de présentation. Ainsi, en cycle 1, si les élèves deviennent progressivement capables d'utiliser les représentations proposées par l'enseignant pour programmer des trajets de plus en plus élaborés, nous nous interrogeons sur leur compréhension des notions sous-jacentes.

## Médiations instrumentales par l'introduction d'artefacts

Dans les séquences observées du cycle 1 au cycle 3, nous constatons que de multiples artefacts matériels et symboliques sont mobilisés au cours de l'activité (figure 4–figure 6) qui constituent un système d'instruments (Bourmaud et Rabardel, 2005) mobilisé pour atteindre l'objectif visé. Aussi, nous avons porté une attention particulière aux fonctions assurées par ces artefacts matériels et symboliques en nous interrogeant sur leurs fonctions dans le processus de conceptualisation.

En cycle 1, la mise à disposition par l'enseignant d'artefacts qui symbolisent les commandes de contrôle du robot sont privilégiées; ceux-ci remplissent plusieurs fonctions. En les positionnant sur la table à côté du quadrillage, ils permettent à l'enseignant de représenter les pas de déplacement que le robot doit effectuer, ils indiquent également la direction et le sens du déplacement du robot sur le quadrillage. Au moment de la programmation du robot, ils indiquent l'ordre dans lequel les commandes de directions doivent être utilisées. Dans un premier temps, la compréhension de ces symboles et leur utilisation pour programmer est médiée par le langage et par des gestes déictiques. L'enseignant puis certains élèves mettent ainsi en correspondance le symbole « papier », et la touche correspondante sur le BeeBot. Une fois appropriés par les élèves, ces artefacts matériels et symboliques guident l'action des élèves.

Dans les classes observées du CP au CM1, les élèves doivent représenter par eux-mêmes le parcours à effectuer / la séquence d'instructions à programmer, en utilisant des artefacts dont les propriétés contraignent les représentations que les élèves peuvent construire et les manipulations possibles. Les conventions collectivement partagées incitent également à privilégier l'usage de certaines représentations plutôt que d'autres.

Ainsi, dans la classe de CE1, l'introduction du petit plateau puis de la bande algorithmique contraignent les élèves à schématiser le parcours simulé puis à passer d'une représentation en deux dimensions continue et allocentrée du parcours, à une représentation linéaire, discontinue, normée, et centrée sur l'axe de rotation du robot. Tout au long de cette chaîne de conversions, la représentation matérialisée du parcours perd certaines propriétés pour en gagner d'autres, compatibles avec les normes à utiliser pour programmer. À l'instar de Duval (2017) en didactique des mathématiques, nous considérons que les conversions d'un registre sémiotique à l'autre, en mettant en exergue des propriétés spécifiques du concept travaillé sont constitutives du processus d'abstraction, ici du concept de séquence. Ces représentations sémiotiques sont matérialisées ici par des artefacts, tels que la bande numérique, qui concrétisent les propriétés pertinentes pour la programmation d'une séquence d'instructions.

Dans les différentes séquences observées, les artefacts remplissent ainsi différentes fonctions. Ils sont en premier lieu des traces du parcours / de la séquence. Utilisés collectivement ces artefacts constituent des « objets intermédiaires » (Vinck, 2009) : des supports de négociation au sein des groupes d'élèves avec l'enseignant, qui permettent de préciser les caractéristiques du parcours et les déplacements du robot en introduisant différentes normes et conventions. Ces dernières peuvent être matérialisées par les propriétés des artefacts manipulés : leur taille, leur forme, leur caractère manipulable ou pas, les inscriptions déjà présentes, la labilité des traces. . . Autrement dit, les propriétés des artefacts mis à disposition des élèves (feuilles, plateau, bande numérique, etc.) pour représenter le parcours / le programme contraignent les inscriptions produites par les apprenants ainsi que les conversions d'un registre sémiotique à un autre. Associés à des schèmes de pointage, de comptage et à de multiples manipulations, ces artefacts deviennent un système d'instruments mobilisé pour anticiper le parcours, programmer des déplacements complexes, et déboguer.

Comme le souligne Duval (2017), la conversion d'un registre sémiotique à l'autre se fait spontanément lorsqu'il existe une correspondance sémantique entre unités signifiantes, c'est-à-dire une possibilité de convertir une unité signifiante de départ en une seule unité signifiante dans la représentation d'arrivée. Dès lors,

les représentations sémiotiques matérialisées par ces artefacts peuvent être un soutien ou au contraire un obstacle aux apprentissages. Le choix de ces artefacts matériels et symboliques qui formeront le système d'instrument de l'élève constitue donc un enjeu pédagogique important pour l'enseignant. Il s'agit d'offrir les outils pour représenter distinctement le parcours (souvent continu en deux dimensions) et la séquence d'instructions (discrete et linéaire) en favorisant les conversions d'une représentation à l'autre.

## Conclusion

Le BeeBot – objet programmable conçu sur le modèle de la tortue logo selon une approche constructiviste (Papert, 1980) – suscite chez les élèves de la curiosité et les invite à s'engager dans une activité manipulative ludique. Dans les séquences observées – sous-tendues par une conception socio-constructiviste des apprentissages – les élèves s'engagent collectivement dans une démarche d'investigation qui les amène à se confronter à différents obstacles (qui pourraient être qualifiés d'épistémologiques). Soutenus par l'enseignant et les échanges collectifs, les élèves sont encouragés à dépasser ces obstacles en s'engageant dans une démarche de résolution de problème afin d'en comprendre le fonctionnement et d'apprendre à programmer des déplacements dans l'espace de plus en plus complexes.

Au cours des différentes séances, les élèves sont amenés à travailler leur représentation de l'espace, en réalisant de multiples conversions entre représentation allocentrée du parcours et représentation égocentrée de la séquence d'instructions. Néanmoins, les obstacles rencontrés et les médiations proposées conduisent à des apprentissages différents suivant les classes.

Les multiples obstacles rencontrés (épistémologique et ontogénique) amènent les élèves de cycle 1 à s'interroger sur le fonctionnement mécanique de cet objet technique et à découvrir les commandes de contrôle. Néanmoins, les premiers schèmes mis en œuvre durant cette découverte restent ancrés pendant plusieurs séances (Grugier, 2022). S'ils deviennent progressivement capables de mettre en œuvre une procédure pas à pas pour programmer les déplacements suite à l'introduction de représentations symboliques des parcours par l'enseignant, nous pouvons nous interroger sur leur compréhension des notions informatiques sous-jacentes.

En CP-CE1, les premières investigations conduisent les élèves à se confronter à des obstacles de type épistémologique et ontogénique qui suscitent incompréhensions et interrogations ; ces questionnements permettent d'introduire le caractère différé de l'exécution

du programme et de se mettre collectivement d'accord sur une procédure pour programmer une séquence d'instructions et l'exécuter. Différentes médiations langagières et instrumentales sont ensuite introduites pour que les élèves soient en capacité de planifier et de représenter la séquence d'instructions à programmer. Associés à des gestes de pointage, de comptage et à des rotations, ces artefacts constituent un système d'instrument pour atteindre l'objectif visé. De notre point de vue, la construction progressive de représentations sémiotiques normées et matériellement ancrées pour symboliser la séquence d'instructions participe au processus de conceptualisation de cette notion (Nogry et Spach, 2022). Dès lors, le choix des artefacts matériels et symboliques proposés aux élèves est crucial. Une attention particulière est à porter sur la congruence entre les représentations sémiotiques, et les normes qui sous-tendent la conception du robot.

En CMI, les élèves comprennent très rapidement le fonctionnement du robot, et l'utilisent pour programmer des parcours complexes, construits en utilisant les instruments qui font référence en mathématique, sans rencontrer d'obstacles majeurs. Néanmoins, les caractéristiques du BeeBot, et notamment l'absence d'interface permettant de visualiser les instructions en mémoire, ne facilite pas la correction des erreurs dans l'activité de débogage, ce qui limite les possibilités de construire des séquences complexes ou de travailler d'autres concepts informatiques. Les fonctionnalités réduites de ce robot limitent les possibilités de progression.

Sur le plan pédagogique, outre l'importance pour l'enseignant d'identifier les concepts en jeu pour proposer des situations adaptées, cette synthèse soulève la question du choix des artefacts. En effet, l'enseignement de l'informatique à l'école avec des robots planchers nécessite des compromis, de la part des enseignants, entre les tâches envisagées, les caractéristiques techniques de ces robots et les concepts pouvant être développés avec ces derniers. Par exemple, les robots BeeBot ne permettent pas d'aborder la correction d'un programme existant. De plus, la notion de programme se limite à des représentations graphiques des boutons de commande. Enfin, un moment d'appropriation par de l'observation, de la manipulation, des échanges entre les élèves est, semble-t-il, nécessaire pour ensuite appréhender les concepts en informatique envisageables avec ces robots.

Cette synthèse ouvre par ailleurs sur différentes questions de recherches relatives à l'activité d'apprentissage des élèves. On pourra notamment s'interroger sur la question du transfert des compétences développées avec un robot plancher à d'autres situations de programmation dans lesquelles des langages dédiés sont utilisés. Elle ouvre également des questions relatives aux pratiques et à l'activité des enseignants ; une étude systématique de l'activité de préparation des séances pourrait permettre de comprendre

comment ceux-ci se familiarisent avec les concepts travaillés et conçoivent l'instrumentation de ces séquences d'apprentissage en vue de leur proposer des ressources adaptées.

L'utilisation du robot BeeBot en classe est l'une des modalités pédagogiques développées pour introduire l'enseignement-apprentissage de l'informatique à l'école primaire. Cet objet, considéré comme facile d'utilisation, est présent dès la maternelle dans des démarches d'investigation pour proposer des activités de codage. Pourtant son utilisation ne va pas sans poser des difficultés, notamment chez les élèves les plus jeunes.

À partir de l'analyse de l'activité des élèves dans des classes de cycle 1, 2 et 3, il s'agit de rendre compte des processus d'appropriation de ces objets programmables dans ces différents cycles. Une analyse des genèses instrumentales à l'œuvre dans différentes séquences d'apprentissage permet de rendre compte des principales difficultés rencontrées – telles que le manque de compréhension du caractère différé de l'exécution du programme et la non-perception du stockage des instructions dans une mémoire – ainsi que des différentes médiations proposées pour favoriser la construction de schèmes rendant possible la programmation effective de ces robots.

## Recommandations

Le BeeBot est un artefact adapté pour proposer aux élèves une démarche d'investigation, reposant par exemple sur la résolution de problèmes. D'apparence simple, il fait néanmoins apparaître un certain nombre d'obstacles ; des médiations doivent être proposées pour conduire les élèves à s'approprier cet artefact, et à comprendre les concepts sous-jacents (mémoire, programme, séquence d'instruction, etc.).

Voici quelques points de repère pour structurer cette démarche :

- L'investigation du fonctionnement du robot peut se faire par une analyse fonctionnelle en identifiant les différents organes de commande et la fonction correspondante par observation systématique des actions que réalise le robot.
- En cycles 1 et 2, la compréhension des commandes d'exécution du programme (touches Go / Clear / Pause) demande à être discutée pendant des moments de regroupements collectifs avec l'enseignant.
- La compréhension de ce qu'est un programme nécessite l'introduction d'un environnement matériel structuré et de différents instruments :
  - l'utilisation de quadrillages permet de matérialiser les pas de déplacement du robot ;
  - et l'usage d'artefacts pour représenter le parcours (schéma / système de flèche / bande algorithmique) et la séquence d'instruction sont nécessaires pour permettre aux élèves de planifier, et de passer d'une représentation en deux dimensions, continue et allocentrée du parcours, à une représentation de la séquence d'instruction linéaire, discontinue, normée, et centrée sur l'axe de rotation du robot).
- En cycle 3, le quadrillage n'est plus nécessaire, les modalités de déplacement du robot sont rapidement comprises, des instruments mathématiques peuvent être introduits (règles, calculatrices, etc.) permettant la construction d'un algorithme structuré en amont de la programmation.

## Références

- Baron, G.L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie*, vol. 195, n° 2, p. 51–62. <https://doi.org/10.4000/rfp.5032>
- Benitti, F.B.V. (2012). Exploring the educational potential of robotics in schools : A systematic review, *Computers and Education*, vol. 58, n° 3, p. 978–988. <https://doi.org/10.1016/j.compedu.2011.10.006>
- Bers, M.U., Flannery, L., Kazakoff, E.R., et Sullivan, A. (2014). Computational thinking and tinkering : Exploration of an early childhood robotics curriculum, *Computers and Education*, vol. 72, p. 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>

Bisault, J., Le Bourgeois, R., Thémines, J.F., Le Mentec, M., et Chauvet-Chanoine, C. (2023). *Objets pour apprendre, objets à apprendre. Quelles pratiques enseignantes pour quels enjeux?*, ISTE group.

Bourmaud, G., et Rabardel, P. (2005). « Instruments et systèmes d'instruments », dans P. Rabardel, P. Pastré (dir.), *Modèles du sujet pour la conception. Dialectiques activités développement*, Octarès.

Braitenberg, V. (1984). *Vehicles : Experiments in synthetic psychology*, MIT Press.

Duval, R. (2017). *Understanding the mathematical way of thinking—The registers of semiotic representations*, Springer. <https://doi.org/10.1007/978-3-319-56910-9>

Greff, E. (1999). En quoi le robot Algor constitue-t-il un objet didactique original?, *Revue de l'enseignement public et informatique*, vol. 93, p. 127-150.

Grugier, O. (2020). Éducation technologique dans des classes de maternelle. Apprentissages premiers dans l'utilisation et la compréhension d'un artefact robotisé, *Recherche en didactique des sciences et des technologies*, vol. 22, p. 61-92. <https://doi.org/10.4000/rdst.3328>

Grugier, O. (2021). Manipulations de robots programmables en classe par des élèves de 9-10 ans. Éducation au numérique et culture technique, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 28, n° 3, p. 71-94. <https://hal.science/hal-03671165v1>

Grugier, O., et Nogry, S. (2022). « Professional testimony : a programmable object for learning computer science at elementary school », dans J. Bisault, R. Le Bourgeois, J.-F. Thémines, M. Le Mentec et C. Chauvet-Chanoine. (dir.), *Objects to learn about and objects for learning : which practices for which issues?*, ISTE group, p. 159-174. <https://doi.org/10.1002/9781119902171.ch8>

Kazakoff, E.R., Sullivan, A., et Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, vol. 41, n° 4, p. 245–255. <https://doi.org/10.1007/s10643-012-0554-5>

Komis, V. et Misirli A. (2015). « Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables », dans Drot-Delange B., Baron G.-L. et Bruillard E. (dir.), *Informatique en éducation : perspectives curriculaires et didactiques*. Presses universitaires Blaise-Pascal, p.209-226.

Komis, V., Romero, M., et Misirli, A. (2017). « A scenario-based approach for designing educational robotics activities for co-creative problem solving », dans D. Alimisis, M. Moro, et E.



Menegatti (dir.), *Educational Robotics in the Makers Era*, Springer, p. 158-169. <https://doi.org/10.1007/s10643-012-0554-5>

Lebeaume, J. (2019). Objets puis systèmes techniques au programme : éclairages pour une discussion de leurs statuts et de leurs fonctions dans l'enseignement, *Recherches en didactiques*, vol. 27, p. 13-24. <https://doi.org/10.3917/rdid.027.0011>

Levy, S. T., et Mioduser, D. (2008). Does it “want” or “was it programmed to...”? Kindergarten children’s explanations of an autonomous robot’s adaptive functioning, *International Journal of Technology and Design Education*, vol. 18, n° 4, p. 337-359. <https://doi.org/10.1007/s10798-007-9032-6>

Nogry, S., et Spach, M. (2022). Robotics learning at elementary school : Constructing abstractions using multiple instruments, *Frontiers in Education Technology*, vol. 5, n° 2, p. 36. <https://doi.org/10.22158/fet.v5n2p36>

Papert, S. (1980). *Mindstorms : Children, Computers, and Powerful Ideas*, Basic Books.

Rabardel, P. (1995). *Les hommes et les technologies, Approche cognitive des instruments contemporains*, Armand Colin.

Rogalski, J. (2015). Psychologie de la programmation, didactique de l'informatique. Déjà une histoire. . ., *Informatique en éducation : perspectives curriculaires et didactiques*.

Roy, A. (2015). Les fonctions exécutives chez l'enfant : des considérations développementales et cliniques à la réalité scolaire, *Dev*, vol. 7, p. 13-40.

Spach, M. (2017). *Activités robotiques à l'école primaire et apprentissage de concepts informatiques : quelle place du scénario pédagogique ? Les limites du co-apprentissage*, thèse de l'université Sorbonne Paris Cité. <https://theses.hal.science/tel-02271924>

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M. et Yeo, S. H. (2016). A review on the use of robots in education and young children, *Journal of Educational Technology & Society*, vol. 19, n° 2, p. 148-163.

Vinck, D. (2009). De l'objet intermédiaire à l'objet frontière. Vers la prise en compte du travail d'équipement, *Revue anthropologique des connaissances*, vol. 3, n° 1, p. 51-72.

Pour citer ce chapitre :

Nogry, Sandra, et Grugier, Olivier (2024). « Usages des robots programmables BeeBot en classe : des objets pour apprendre, des objets à apprendre », dans Cédric Fluckiger, Laetitia Boulc’h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 115-134. <https://doi.org/10.53480/2024iecare06t/>

# Zoom sur quelques erreurs récurrentes lors des premiers apprentissages en algorithmique

# 7

Marielle LÉONARD<sup>1,2</sup>

1. Université de Lille, ULR 4354, CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, 59000 Lille, France

2. Université de Lille, CNRS, Centrale Lille, UMR 9189 CRISTAL, 59000 Lille, France

L'apprentissage des bases de l'algorithmique est abordé à l'école obligatoire notamment en utilisant des environnements de programmation par blocs comme Scratch. Une des approches possibles pour cet apprentissage est la résolution de puzzles, c'est-à-dire de problèmes fermés dont la tâche est prescrite et dont l'évaluation est effectuée automatiquement par le système, déterminant si chaque solution soumise est valide ou non (Pelánek et Effenberger, 2022). Nous étudions les premiers apprentissages en algorithmique à travers la confrontation d'élèves de 7 à 15 ans avec des puzzles de programmation impliquant de commander un robot virtuel sur une grille, en langage Scratch.

L'enjeu de cette contribution est d'identifier des erreurs récurrentes et leur conceptualisation sous-jacente lors des premiers apprentissages en algorithmique, afin notamment de fournir des repères conceptuels aux enseignants en charge de l'initiation à ce domaine. Notre démarche repose sur la collecte de traces lors de l'activité des élèves dans l'environnement de programmation, et sur un processus d'analyse de ces traces d'interaction associant traitement automatisé et travail d'expert ancré dans le cadre d'analyse de la théorie des champs conceptuels (Vergnaud, 1991). Ce cadre théorique, qui sera présenté plus en détails dans la section « Cadre théorique », vise à comprendre la conceptualisation notamment dans le cas des activités cognitives complexes dont la programmation informatique fait partie. Une difficulté de ce travail d'analyse concerne le volume important des traces collectées, qui exclut une exploration exhaustive à la main. Il est nécessaire de se doter d'une stratégie pour combiner analyse qualitative d'expert et automatisation du traitement de données. Nous présentons cette stratégie et montrons le type de résultat qu'il est possible d'obtenir, en prenant un exemple représentatif qui traite de la notion de boucle (figure 1).

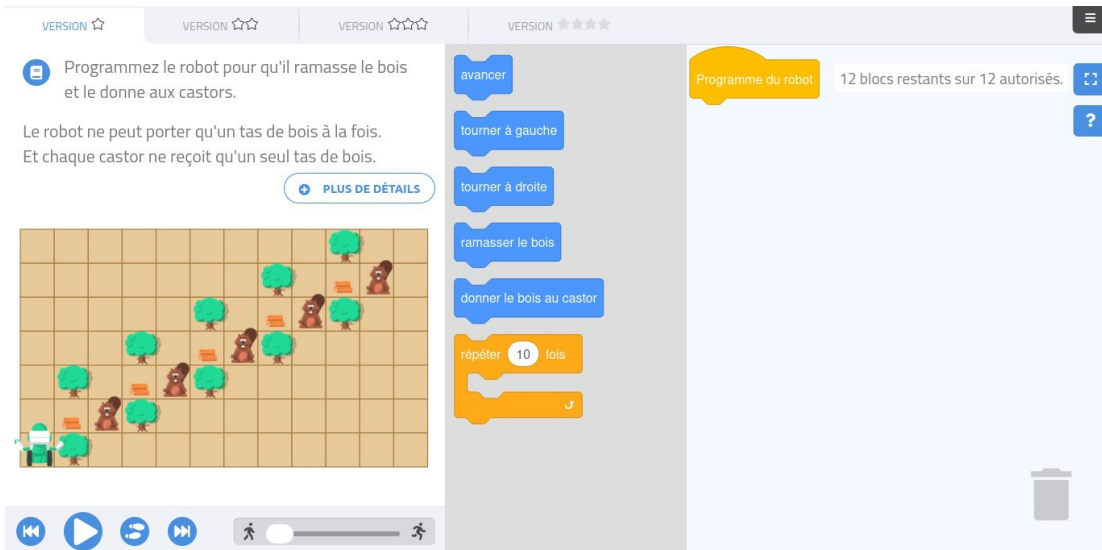


FIGURE 1 – Exemple de puzzle de programmation en langage Scratch dans l'environnement Algoréa : « Distribuer le bois aux castors » en version 1 étoile. © France-ioi.

## Cadre théorique

Pour notre étude sur les premiers apprentissages en algorithmique, nous cherchons à nous placer dans un cadre d'analyse qui nous permette d'interpréter les erreurs des élèves et de comprendre leur raisonnement. La théorie des champs conceptuels élaborée par Vergnaud dans le but de comprendre la conceptualisation dans le cas des activités cognitives complexes répond à ce besoin. Cette théorie s'inscrit dans une approche constructiviste et cognitiviste de l'apprentissage. Vergnaud reprend dans ses travaux une large partie de l'approche constructiviste de Piaget, notamment le concept de schème et la théorie de l'adaptation (assimilation, accommodation, équilibration) (Piaget, 1935). En prenant pour unité d'analyse le couple sujet/situation, Vergnaud étudie le sujet lorsqu'il se trouve dans une situation où son but est de réaliser une tâche. Il fait l'hypothèse que toute action finalisée repose sur une « conceptualisation-en-acte », c'est-à-dire que les actions du sujet révèlent son activité cognitive sous-jacente, quand bien même celle-ci n'est pas verbalisée (Vergnaud, 1991).

Dans notre contexte, la tâche finalisée de l'élève est de concevoir un programme en langage Scratch pour que le robot virtuel effectue une certaine mission, formulée dans l'énoncé et évaluée automatiquement lors de l'exécution du programme. Nous faisons l'hypothèse que les traces d'interaction collectées rendent

compte d'une part significative des actions de l'élève sur l'interface, et sont de ce fait révélatrices de ce que Vergnaud nomme conceptualisation-en-acte.

Afin de préciser comment opère cette conceptualisation-en-acte, Vergnaud développe deux axes. Un premier axe est porté par les situations. Le second est porté par le sujet en action, à travers le concept de schème.

Dans le but de comprendre l'activité du sujet, Vergnaud invite à mener en premier lieu une analyse épistémologique des situations auxquelles le sujet est confronté. Cette analyse amène à regrouper les situations singulières en classes de situations suivant la manière dont elles sont traitées par le sujet. Il s'agit d'identifier des paramètres, appelés « variables de situation », qui permettent de différencier des situations proches. Deux cas se présentent suivant que le changement de valeur d'une variable de situation affecte ou non la structure du traitement de la situation par le sujet. Dans le cas où le sujet traite les deux situations de façon similaire, celles-ci appartiennent à la même classe. Dans le cas contraire, deux classes de situations sont distinguées en fonction de la valeur de la variable de situation.

Pour traiter une situation à laquelle il est confronté, le sujet mobilise des schèmes, qui sont des unités d'action déjà prêtes. Vergnaud approfondit le concept de schème introduit par Piaget en en donnant une définition analytique.

Composantes d'un schème selon Vergnaud :

- un but qui est la finalité de l'action ;
- des règles de conduite de l'action, qui se succèdent (règles de prise d'information, règles d'action proprement dite, règles de contrôle de l'action) ;
- des invariants opératoires de deux types, qui constituent la partie cognitive du schème :
  - « concept-en-acte » qui est « un concept tenu pour pertinent dans l'action en situation » (Vergnaud, 2013) ;
  - « théorème-en-acte » qui est « une proposition tenue pour vraie dans l'action en situation » (Vergnaud, 2013) ;
- des inférences, c'est-à-dire des raisonnements en situation qui visent à prendre en compte les spécificités d'une situation et à anticiper les résultats de l'action.

À titre d'exemple, nous renseignons de manière analytique le schème de dénombrement mentionné plusieurs fois dans les travaux de Vergnaud (1991, 2001, 2007).

### Schème de dénombrement

- But : « Associer un nombre à une collection discrète » (Vergnaud, 2001).
- Règles de conduite de l'action.
  - Prise d'information : en amont de l'action, identifier les éléments à compter, repérer quelle est l'unité.
  - Action proprement dite : pointer les unités les unes après les autres en leur associant un mot nombre.
  - Contrôle : contrôle au cours de l'action de la coordination entre le regard, le geste de pointage et l'attribution du mot nombre, recomptage éventuel à titre de vérification.
- Invariants opératoires dont certains correspondent aux principes de Gelman (Gelman et Gallistel, 1978).
  - Concepts-en-acte : nombre, correspondance terme à terme, cardinal.
  - Théorèmes-en-acte : « le dernier mot prononcé est le cardinal de la collection », « l'ordre dans lequel on considère les éléments est indifférent ».
- Inférences : identifier le prochain élément à traiter ; maintenir deux ensembles : les éléments déjà pointés et ceux qui ne le sont pas encore.

Du point de vue expérimental, Vergnaud construit la théorie des champs conceptuels à partir d'expérimentations à l'échelle de la classe comme celle sur les structures additives en 1976 (Vergnaud et Durand, 1976). Notre objectif est de nous appuyer sur ce cadre théorique mais avec une méthode de collecte et d'analyse de traces d'interaction qui n'était pas envisageable techniquement au moment de l'élaboration de la théorie des champs conceptuels. Notre objectif est, à partir de ces traces d'interaction collectées dans l'environnement de programmation, de considérer les deux axes préconisés par Vergnaud : classes de situations et mobilisation de schèmes.

Dans notre contexte, les situations sont les puzzles de programmation que le sujet a pour but de résoudre. Nous avons déjà mené une analyse épistémologique pour les situations qui requièrent l'utilisation d'une boucle (bloc Scratch « répéter n fois »). Cette analyse a fait l'objet de deux communications (Léonard *et al.*, 2022a ; 2022b). Nous en reprenons les principaux résultats dans la section « Paliers de difficulté observables à large échelle ». Pour ce qui concerne les schèmes mobilisés par le sujet, une partie est

observable à travers les traces d'interaction de celui-ci avec l'environnement de programmation. Une autre partie, comme les gestes que le sujet pourrait effectuer devant sa machine, ne l'est pas dans le cadre de cette expérimentation, et restera inexplorée dans notre travail. Notre étude des schèmes mobilisés par le sujet comporte trois niveaux d'analyse, à des échelles de plus en plus fines. Nous en structurons la présentation à la manière d'un zoom dans les sections « Mobilisation de schèmes révélée par un indicateur de rapidité normalisée », « Zoom sur la procédure experte et les erreurs récurrentes révélées », « Zoom sur la conceptualisation-en-acte du sujet ». Pour chaque étape, nous présentons quelques résultats obtenus en nous appuyant sur un exemple « fil rouge » (figure 1).

## Cadre expérimental et méthodologie

Les puzzles de programmation qui constituent notre cadre expérimental sont issus de l'édition 2022 du concours de programmation Algoréa<sup>1</sup>. La participation au concours Algoréa est organisée en milieu scolaire, sous la forme de sessions en temps limité, d'une durée de 45 minutes. Les élèves progressent dans des catégories de couleur en fonction de leurs résultats. Pour la présente étude, nous ne considérons que les deux catégories les plus faciles (blanche et jaune), dans lesquelles les notions algorithmiques de base sont abordées : séquence d'instructions, répétition, structure conditionnelle, variable et fonction. Les parcours dans ces catégories sont composés de six problèmes. Chaque problème comporte quatre versions de difficulté croissante repérée par des étoiles, de la version 1 étoile qui est la plus facile à la version quatre étoiles qui est la plus difficile. Nous avons donc 24 puzzles par parcours, chaque puzzle correspondant à une situation au sens de Vergnaud. Toutes les situations consistent à programmer des actions et des déplacements d'un robot virtuel sur une grille.

L'objectif est de disposer, pour les mêmes situations, à la fois de données globales sur la population totale et de données locales selon la distinction de Pelanek (*global data / local data*) (Pelánek, 2017). Dans cette optique, nous collectons des données à trois échelles différentes.

- À l'échelle nationale : échantillon global des participations individuelles au concours Algoréa en langage Scratch – de 7800 à 92 400 élèves de 9 à 15 ans suivant la catégorie et le tour. Par exemple, 36 924 élèves ont été confrontés au puzzle « Distribuer le bois au castor » en version 1 étoile de la figure 1.
- À l'échelle de classes : neuf classes de collège (deux 6<sup>e</sup>, trois 5<sup>e</sup>, un 4<sup>e</sup>, trois 3<sup>e</sup>) – 252 élèves de 11 à 15 ans – qui ont

1. Site du concours Algoréa : <https://algorea.org>

participé sur une année (deux ou trois sessions par élève). À cette échelle, nous disposons de l'enregistrement de deux types d'évènement horodaté.

- Lorsqu'un élève lance le programme qu'il a conçu avec des blocs Scratch, celui-ci est transcrit et enregistré dans un format textuel (xml). Pour le puzzle de la figure 1, cela représente 1099 programmes collectés.
- Les changements de puzzles sont aussi enregistrés, ce qui permet de calculer la durée passée par l'élève sur chaque puzzle.
- À l'échelle du sujet : 20 élèves de 7 à 15 ans (CE1 à la 3<sup>e</sup>). Pour chacun de ces sujets, nous disposons de l'enregistrement vidéo des trois sessions de l'année 2022 (enregistrement de l'écran et de la voix par Zoom) en plus des traces d'interaction comme à l'échelle précédente.

Ces choix méthodologiques se justifient par la consultation de travaux sur l'analyse de traces d'interaction.

Un premier point concerne le niveau de granularité des données collectées. Parmi les niveaux de granularité (du plus grossier au plus fin) identifiés par Ihantola *et al.* (2015), la collecte de tous les programmes exécutés correspond au deuxième niveau sur six. Ce niveau de granularité est un compromis entre une granularité assez fine pour mener les analyses prévues d'une part, et un volume de stockage et une capacité de traitement de ces données d'autre part. En effet, collecter seulement le programme final (niveau de granularité le plus grossier) contraint à inférer des bribes de processus à partir de celui-ci. En revanche, collecter les programmes intermédiaires permet de rendre compte plus précisément du processus de résolution et des erreurs commises, ce qui est notre objectif. En ce qui concerne la collecte du flux continu d'activité par enregistrement vidéo de l'écran de l'utilisateur, elle ne fait pas partie des niveaux de granularité de Ihantola *et al.* car les auteurs n'ont répertorié que la collecte de données discrètes (évènements).

Un second point est relatif au format de collecte des programmes édités. Si des captures d'écran à partir de sessions enregistrées en vidéo peuvent être envisagées pour des analyses qualitatives à l'échelle réduite de quelques sujets comme dans l'étude de Fernandez *et al.* (2022), des analyses quantitatives à plus large échelle imposent un format textuel de collecte. Pelánek et Effenberger (2022) listent plusieurs possibilités de format textuel pour la collecte des programmes de l'utilisateur, parmi lesquels figure le format xml. Si l'enregistrement des programmes dans un format textuel permet des traitements à large échelle, il implique aussi de se doter d'outils pour régénérer les images afin d'être en mesure

de construire des visualisations facilement interprétables, ce que nous avons fait.

Ainsi, en mettant en place cette méthodologie de collecte et d'analyse de données à trois échelles différentes, nous visons à la fois des analyses très précises et une robustesse statistique en établissant une correspondance entre les analyses à ces trois échelles. Nous répartissons les données collectées de la manière suivante par rapport aux deux axes d'étude de la théorie des champs conceptuels.

- Les données relatives aux puzzles et les données décrivant globalement l'ensemble de la population confrontée à ces puzzles renseignent l'axe des situations.
- Les données locales renseignent l'axe des schèmes, de manière d'autant plus précise que la granularité est fine. L'interprétation de ces données devrait nous permettre d'identifier et de documenter les différentes composantes d'un schème présentes dans la définition analytique qu'en donne Vergnaud (voir la section « Cadre théorique »).

## Paliers de difficulté observables à large échelle

À l'échelle large de l'ensemble des participations individuelles au concours Algoréa en langage Scratch, nous disposons du taux de réussite pour chaque puzzle et chaque niveau de classe. C'est à cette échelle que nous menons une analyse épistémologique des situations comme le préconise Vergnaud. Par une analyse *a priori* des situations, nous identifions des variables qui permettent de distinguer des classes de situations. Les taux de réussite nous aident en ce sens dans la mesure où des puzzles proches dont le taux de réussite est significativement différent ne sont très probablement pas appréhendés de la même manière par le sujet, sans que l'on puisse à cette échelle déterminer ce qui se passe du point de vue de l'activité du sujet.

Nous reprenons les puzzles pour lesquels nous avons déjà publié une étude et dont la résolution implique d'utiliser une boucle. Pour ces puzzles, nous avons identifié des classes de situations en lien avec l'identification de motifs, habileté qui relève de la pensée informatique (Csizmadia *et al.*, 2015). Nous avons défini un « motif » comme une entité repérable au sein d'un ensemble car répétée à l'identique ou avec des variations prédictibles. Cette définition nous a amenés à distinguer deux types de motifs dans le contexte de la programmation d'un robot virtuel sur une grille. D'une part, un « motif visuel », constitué de cases adjacentes, est observable sur la grille. D'autre part, un « motif algorithmique », constitué



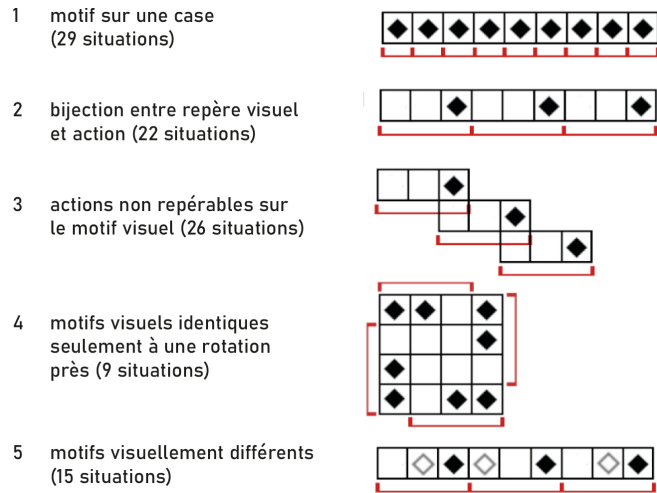


FIGURE 2 – Classes de situations relatives à l'identification de motifs dans un contexte de programmation d'un robot virtuel sur une grille.

d'une séquence d'actions que le robot répète dans le même ordre chronologique, n'est observable que lors de l'exécution effective de ces actions (Léonard *et al.*, 2022b).

Dans un programme conçu en langage Scratch, le motif algorithmique correspond à la séquence de blocs dans le corps de la boucle. Les classes de situations que nous avons identifiées constituent une gradation dans la difficulté de ces puzzles, qui dépend des caractéristiques du motif visuel et du degré de correspondance entre motif visuel et motif algorithmique. La figure 2 montre cette gradation. Les motifs, constitués de cases adjacentes, y sont délimités par les repères en rouge.

Dans la suite de cette contribution, nous nous concentrons sur la troisième classe de situations : le motif visuel s'étend sur plusieurs cases de la grille, les motifs occupent les deux dimensions de la grille, le système d'orientation du robot est relatif. Dans ces situations, certaines actions du robot (les actions de pivotement) ne sont pas directement repérables sur le motif visuel.

Le puzzle « Distribuer le bois aux castors » en version 1 étoile de la figure 1 est une situation singulière relevant de cette classe de situations. Nous construisons la courbe des taux de réussite relevés à l'échelle nationale pour ce puzzle sur les différents niveaux de classe et le mettons en regard des résultats pour la classe de situations dans laquelle il est catégorisé (figure 3).

Dans cette classe de situation, le puzzle choisi est une des instances les plus faciles. Son taux de réussite s'échelonne entre 70 % en classe de cours moyen 1<sup>re</sup> année (CM1) et 90 % en classe de 3<sup>e</sup>. Il est situé en-dehors de la zone interquartile (zone qui comprend la moitié des puzzles appartenant à cette classe de situations). En effet, plusieurs difficultés mises en évidence par ailleurs ne se

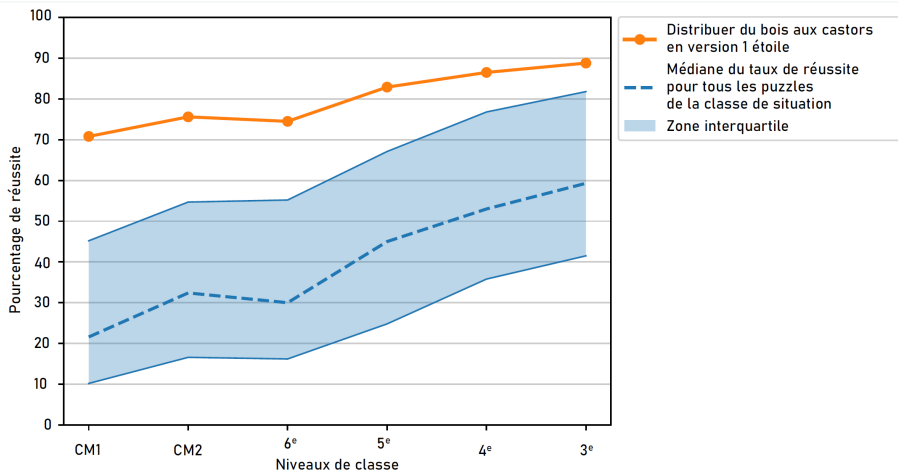


FIGURE 3 – Taux de réussite pour le puzzle « Distribuer le bois aux castors » version 1 étoile en fonction des niveaux de classe.

retrouvent pas dans cette situation : boucle unique *vs.* séquence de boucles, présence de points de repère concernant le motif (bois, castors, disposition des arbres qui guide le parcours du robot), pas d'instructions avant la boucle, pas de cases adjacentes identiques qui appartiennent au même motif. En conséquence, cette situation devrait faciliter l'identification des difficultés spécifiques à cette classe de situations.

## Mobilisation de schèmes révélée par un indicateur de rapidité normalisée

Dans cette section, nous étudions la mobilisation de schèmes de manière quantitative, en analysant les exécutions de programme et leur validité. La difficulté d'un puzzle se traduit dans les traces d'interaction par une augmentation du nombre d'essais et du temps passé sur la tâche. Nous mettons en correspondance la variation de la valeur de ces indicateurs avec deux cas dans lesquels peut se trouver le sujet face à une situation, qui sont décrits dans de la théorie des champs conceptuels : un premier cas où le sujet mobilise un schème opérationnel pour traiter la situation de manière experte en mobilisant un schème déjà prêt et un second cas où le sujet entre dans un processus d'accommodation où il remet en cause le système de schèmes précédemment établi.

Afin d'illustrer le propos, nous proposons de partir de la visualisation de la session d'un sujet singulier sous forme de frise chronologique (figure 4). Ce type de visualisation, qui met en évidence l'aspect temporel de l'activité du sujet, est répertorié parmi les techniques de visualisation (Khan et Khan, 2011). Il a

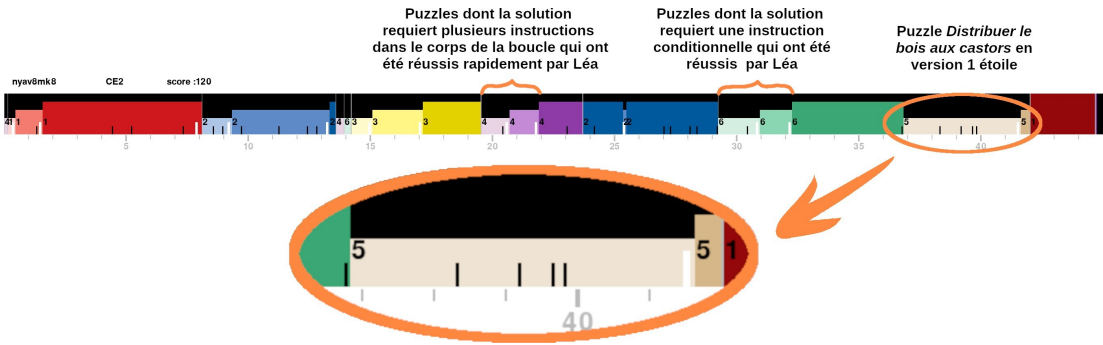


FIGURE 4 – Visualisation d'une session d'élève sous forme de frise chronologique, avec zoom sur le puzzle « Distribuer le bois aux castors » en version 1 étoile.

déjà été mobilisé dans le contexte de la programmation par blocs (Fernandez *et al.*, 2022).

## 2. Le prénom a été changé.

Les problèmes du parcours sont numérotés de 1 à 6. Chaque problème est représenté par une couleur dédiée. Chaque version est représentée par une hauteur et une nuance, des versions 1 étoile, les plus claires et plus basses, jusqu'aux versions 4 étoiles, les plus foncées et plus hautes. Léa<sup>2</sup> est élève de cours élémentaire 2<sup>e</sup> année (CE2). Elle fait partie des élèves dont les sessions ont été enregistrées en vidéo. Léa a abordé le puzzle 5 « Distribuer le bois aux castors » en version 1 étoile de la figure 1 en fin de session. Auparavant, elle a validé le problème 4 en versions 1 et 2 étoiles assez facilement, ce qui signifie qu'elle a franchi le palier du passage de une à plusieurs instructions dans le corps de la boucle. Elle possède aussi une première maîtrise de l'instruction conditionnelle (bloc « si/sinon » en Scratch) qui est mise en évidence par la validation des versions 1 et 2 étoiles du problème 6. Pour le puzzle « Distribuer le bois aux castors » en version 1 étoile dont nous détaillons l'analyse, nous observons que plusieurs essais et une durée de près de 5 minutes ont été nécessaires à Léa pour réussir le puzzle. La procédure n'est donc pas experte, Léa est très probablement entrée dans un processus d'accommodation qui a abouti positivement, ce que nous argumenterons dans les sections suivantes.

Au-delà de l'analyse de frises chronologiques de sujets singuliers, nous cherchons à réaliser une analyse quantitative de l'ensemble de nos données à l'échelle intermédiaire des classes.

À cette fin, nous nous appuyons sur des travaux de recherche relatifs à la construction d'indicateurs à partir de traces d'interaction et à l'établissement de profils en rapport avec la résolution d'un puzzle donné. Pelanek définit quatre profils de performance basés sur une combinaison de deux indicateurs : temps passé sur la tâche et catégorisation des erreurs en « commune » et « non

commune » (Pelánek, 2018). Jiang *et al.* (2022) utilisent une méthode de *clustering* (technique de *machine learning* visant à établir des groupements dans des données) afin de caractériser quatre profils à partir des programmes intermédiaires soumis : ceux qui abandonnent rapidement (*quitters*), ceux qui sont loin d'une solution valide lors des premiers essais et s'en approchent plus ou moins sans aboutir (*approachers*), ceux qui sont loin d'une solution valide lors des premiers essais puis s'en approchent jusqu'à trouver (*solvers*), ceux qui soumettent un programme valide dès les premiers essais (*knowers*).

De ces travaux, nous retenons l'idée d'établir des profils fondés sur une combinaison d'indicateurs construits à partir des traces d'interaction. Dans cette optique, nous construisons un indicateur qui mesure la rapidité avec laquelle un sujet résout un puzzle de programmation, indépendamment de sa dextérité et de la longueur de la solution de référence. Cet indicateur de rapidité est normalisé, c'est-à-dire qu'il est rendu indépendant de la vitesse du sujet pour placer un bloc dans l'éditeur et du nombre de blocs à déplacer pour obtenir la solution de référence. Ainsi, il permet de comparer des résolutions entre elles et de procéder à des traitements statistiques tout en rendant compte, au niveau individuel, de la durée irréductible à l'exécution de la tâche et du temps éventuellement nécessaire à un processus d'accommodation.

Lorsque le sujet a résolu un puzzle, une valeur basse (voire négative) de l'indicateur de rapidité normalisée signifie que cette résolution a été aisée. À l'inverse, une valeur élevée de l'indicateur de rapidité normalisée, souvent concomitante à un nombre d'essais élevé lui aussi, signifie que le sujet a rencontré des difficultés lors de la résolution du puzzle. Pour une explication détaillée de la construction de cet indicateur, nous invitons le lecteur à consulter l'article *Détermination de profils relatifs à la mobilisation de schèmes lors de la résolution de puzzles de programmation* (Léonard *et al.*, 2023). Notez que le nom de l'indicateur a été changé pour plus de clarté, mais la construction reste absolument inchangée.

À partir de la combinaison de cet indicateur de rapidité normalisée et du nombre d'essais, nous établissons des profils qui caractérisent le type de résolution pour un couple sujet/puzzle.

- Expert : la résolution du puzzle est experte, la réussite ne nécessite qu'un seul essai, et l'indicateur de rapidité normalisée est inférieur à 0,5.
- Ajustement : la réussite intervient en deux ou trois essais, avec un indicateur de rapidité normalisée inférieur à 0,5.
- Accommodation : la réussite intervient après une résolution qui est marquée par un indicateur de rapidité normalisée supérieur à 0,5 ou un nombre d'essais strictement supérieur à trois (ou les deux).

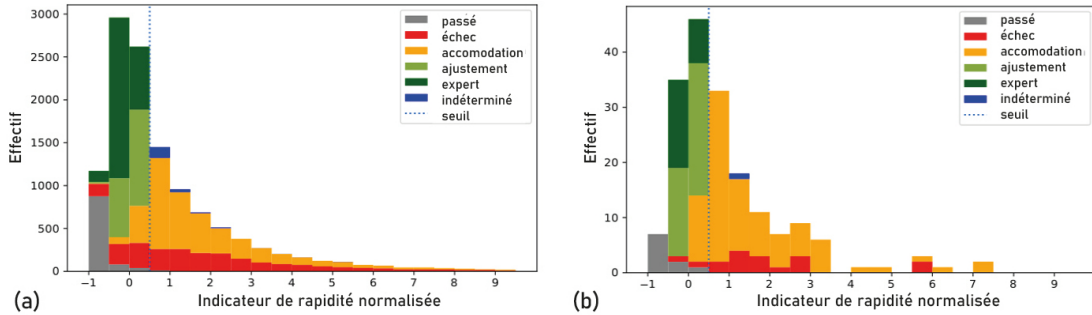


FIGURE 5 – Histogrammes des profils de résolution. (a) Tous les puzzles des catégories blanche et jaune du concours Algoréa 2022. (b) Puzzle « Distribuer le bois aux castors » en version 1 étoile. Source : Léonard *et al.* (2023).

- Échec : tous les programmes exécutés sont invalides.
- Passé : le puzzle a été ouvert mais aucun programme n'a été exécuté.
- Indéterminé : le type de résolution ne peut pas être déterminé (réussite en un seul essai avec un indicateur de rapidité normalisée supérieur à 0,5).

Nous reprenons ici l'histogramme des profils pour toutes les confrontations sujet/puzzle des élèves des neuf classes de collège de notre étude en 2022 (figure 5a). Nous montrons (figure 5b), la répartition des profils pour les seules confrontations avec le puzzle « Distribuer le bois aux castors » pour ces mêmes élèves et les 20 élèves dont la session a été enregistrée en vidéo.

Nous observons que l'allure générale du diagramme en barres est conservée. Cependant, la répartition des profils est sensiblement différente pour ce puzzle.

Le profil accommodation est très répandu pour le puzzle « Distribuer le bois aux castors » version 1 étoile (49,5 % contre 31,5 % pour l'ensemble des puzzles) alors que les profils expert et échec le sont moins. Nous déduisons de cette répartition particulière que ce puzzle est porteur d'une difficulté qui engendre un processus d'accommodation pour près de la moitié des sujets. Cette difficulté est surmontée pendant la résolution car le taux d'échec est bas (9,3 % pour notre échantillon). Notre objectif est d'identifier la nature de cette difficulté : erreurs commises, conceptualisation erronée.

Si nous revenons à l'exemple de Léa, nous relevons que cette élève, avec cinq essais et un indicateur de rapidité normalisée de 0,82 a un profil accommodation pour ce puzzle. Son indicateur de rapidité normalisée se situe entre 0,5 et 1, la tranche la plus répandue pour le profil accommodation, ce qui nous incite à étudier son activité face à ce puzzle de manière plus approfondie.

Profils	Ensemble des puzzles	Puzzle « Distribuer le bois aux castors » version 1 étoile
Expert	23,4	13,2
Ajustement	15,5	22,0
Accommodation	31,3	49,5
Échec	19,5	9,3
Passé	8,6	5,5
Indéterminé	1,6	0,5

**TABEAU 1** – Répartition des profils de résolution (en pourcentage).

## Zoom sur la procédure experte et les erreurs récurrentes révélées

Nous considérons dans cette section l'ensemble des programmes soumis par les élèves sur le puzzle « Distribuer le bois aux castors » version 1 étoile déjà abordé précédemment.

La figure 6 montre les programmes non valides soumis par les élèves sur ce puzzle. Ces programmes sont triés par ordre décroissant d'occurrences.

Le programme incorrect le plus souvent soumis est un programme dans lequel il manque le bloc « tourner à gauche » en fin de boucle. Plus d'un tiers des élèves (34 %) soumettent ce programme erroné. L'erreur commise est le non remplacement du robot relativement au motif visuel qu'il doit parcourir lors de l'itération suivante. En utilisant des expressions régulières, nous cherchons tous les programmes qui comportent une erreur de ce type. Pour 52 % des sujets (89 sur les 172 qui ont soumis au moins un programme pour ce puzzle), cette erreur de non remplacement du robot relativement au motif visuel est présente dans au moins l'un des programmes exécutés. Concernant les neuf classes de collège qui ont participé à notre étude, 46 % des élèves de 6<sup>e</sup>, 45 % des élèves de 5<sup>e</sup>, 61 % des élèves de 4<sup>e</sup> et 57 % des élèves de 3<sup>e</sup>, ont commis cette erreur. Elle est donc très répandue et elle ne régresse pas avec l'âge des élèves au collège.

Nous regardons à présent quel est le profil de résolution des élèves qui ont commis cette erreur. Pour une part significative des élèves, la soumission de ce programme va de pair avec un échec dans la résolution du puzzle, ce qui n'est pas le cas par exemple pour le programme d'id 742 qui est le deuxième plus fréquemment soumis. Dans ce dernier cas, l'erreur concerne le nombre d'itérations à renseigner dans le bloc « répéter ». Ce nombre est laissé à sa valeur par défaut, mais il est ensuite rectifié dans tous les cas. En revanche, l'absence du bloc « tourner à gauche » correspond à une difficulté conceptuelle récurrente, que nous proposons de creuser.

**PROGRAMMES NON VALIDES**

Nb occur.	70	30	26	25	20	20
Freq. sujets	0.34	0.17	0.14	0.12	0.12	0.09
Ajustement	0.36	0.2	0.12	0	0.2	0.07
Accommodation	0.53	0.8	0.71	0.57	0.55	0.6
Échec	0.12	0	0.17	0.43	0.25	0.33

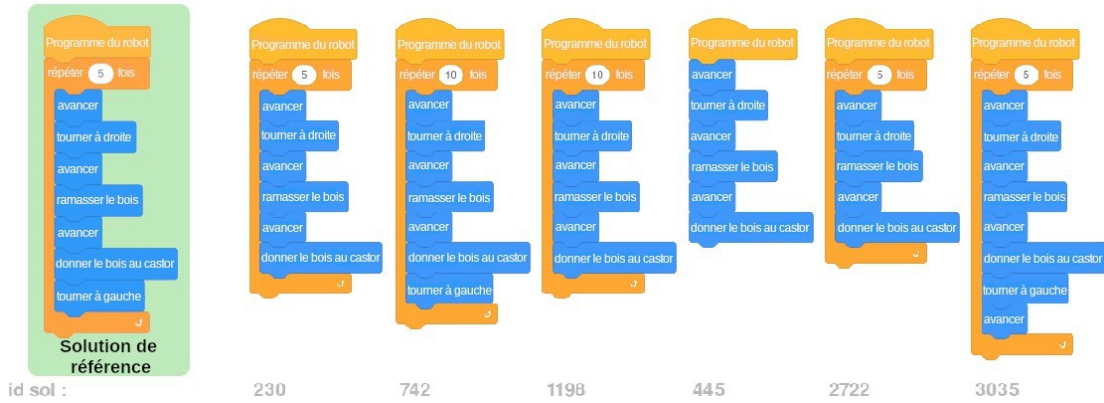


FIGURE 6 – Programmes non valides soumis par ordre décroissant d’occurrences. Crédit : France-ioi.

## Zoom sur la conceptualisation-en-acte du sujet

Dans cette section, nous approfondissons l’étude de l’erreur récurrente mise en évidence dans la section précédente : absence du bloc « tourner à gauche » en dernière position de la séquence dans le corps de la boucle. Nous retrouvons cette erreur dans les traces d’interaction de plusieurs élèves qui ont bénéficié de sessions individuelles, certains pendant deux années. Pour ces élèves, nous disposons à la fois d’une visualisation de la succession des programmes soumis pour chaque puzzle et de l’enregistrement vidéo de la session. La succession des programmes soumis, bien qu’ayant une granularité assez fine, reste une collecte de données discrètes. Avec l’enregistrement vidéo, nous passons à un flux continu d’informations du point de vue de la manipulation de l’interface (mouvements de souris, déplacement et suppression de blocs. . .), qui de plus est accompagné des verbalisations du sujet. Nous sommes ainsi mieux en mesure d’identifier les schèmes mobilisés lors de la résolution du puzzle et donc d’appréhender plus finement la conceptualisation-en-acte du sujet.

Nous reprenons l’exemple de Léa et générons la succession des programmes qu’elle a soumis pour le puzzle « Distribuer le bois aux castors » version 1 étoile (figure 7).

La soumission n° 4 de Léa correspond à l’erreur la plus fréquemment commise sur ce puzzle. Nous mettons en relation ces traces d’activité avec l’extrait vidéo correspondant, ce qui nous permet d’accéder au point de vue subjectif du sujet.

Lors de l'ouverture du puzzle, Léa fait part d'une impression de complexité. Lorsque l'expérimentatrice lui demande ce qui lui donne cette impression, sa réponse concerne le nombre d'éléments sur la grille : « Quand on voit tous les trucs là... tous les castors... tous les bois... euh ». Cette première impression est cohérente avec le fait que les puzzles dont la grille est épurée des éléments inutiles sont mieux réussis.

Cela dit, Léa identifie rapidement la classe de situation des problèmes itératifs, même si un manque de dextérité dans le maniement de la souris provoque l'interversion de blocs dans la première soumission. Léa mobilise des schèmes déjà construits pour cette classe de situation, ce que révèlent ses verbalisations.

- Schème d'identification du motif qu'elle se représente comme une marche : « il suffit de répéter plusieurs marches ».
- Schème de dénombrement des motifs : « 1, 2, 3, 4, 5 répéter ».

En revanche, Léa ne voit pas d'emblée l'erreur de non remplacement du robot par rapport au motif visuel. Une fois que le robot a distribué le bois, il a effectué l'ensemble des actions attachées au motif visuel. L'invariance de la position du robot par rapport au motif visuel à parcourir lors de l'itération à venir n'est pas considérée.

Léa a besoin d'une incitation à utiliser le mode pas à pas. Cette exécution des instructions une par une l'amène à prendre conscience de l'erreur après l'exécution de la première itération : « Ah ça y est,

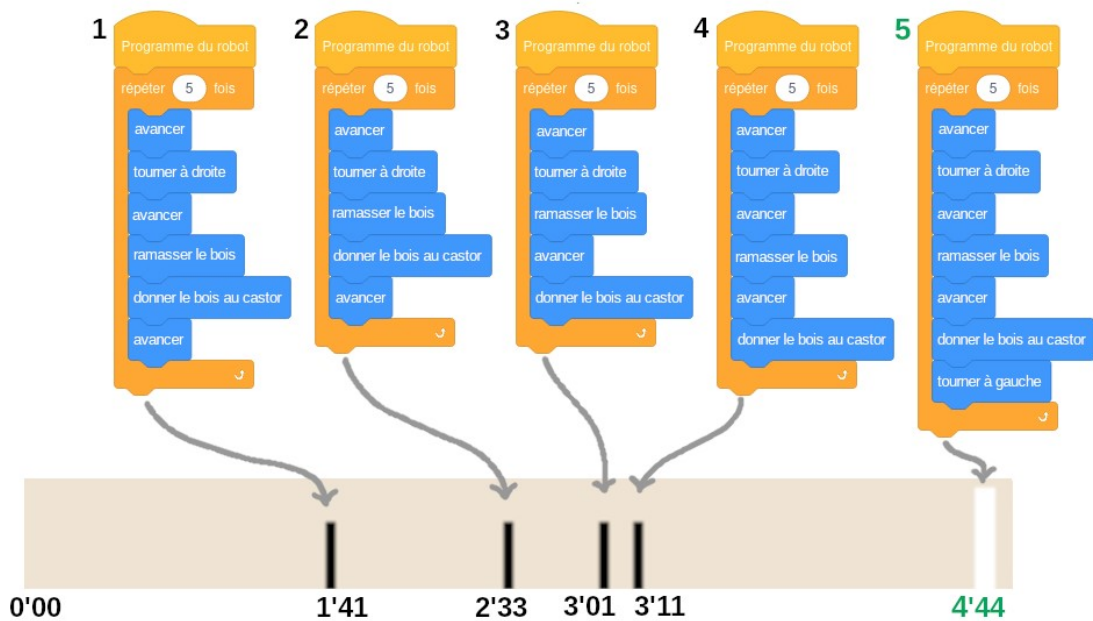


FIGURE 7 – Succession des programmes soumis par Léa sur le problème « Distribuer le bois aux castors » en version 1 étoile. Crédit : France-ioi.



je viens de comprendre ». Sans hésitation, elle place alors le bloc de rotation manquant en dernière position dans le bloc répéter. En même temps, elle donne une explication à son niveau de la cause de l'erreur : « tourner à gauche sinon il va foncer », ce qui prouve qu'une analyse de l'exécution a eu lieu, que la correction n'est pas de l'ordre de la stratégie par essai-erreur, stratégie répandue qui consiste à effectuer des modifications du programme au hasard puis à lancer systématiquement une exécution après chaque modification sans prendre le temps de l'analyse. Par sa remarque, Léa nous donne aussi une indication sur l'élément de l'interface qui l'a aidée à prendre conscience de son erreur : l'arbre dans lequel le robot ne doit pas foncer. L'indication de Léa est cohérente avec des résultats que nous ne détaillons pas dans cette contribution, qui concernent la présence d'éléments sur la grille : ces éléments sont aidants s'ils contraignent le parcours du robot.

Dans le système d'orientation relative, la position du robot comprend la case de la grille sur laquelle il se trouve (ses coordonnées en arrière-plan) et son orientation. Lorsqu'il se représente l'exécution, le sujet doit donc maintenir ces deux informations en mémoire. C'est la gestion de l'orientation du robot qui s'avère être la plus difficile. En effet, à l'inverse de la succession de cases occupées par le robot que le sujet se représente souvent en pointant les cases de la grille avec sa souris, celle-ci n'est pas visualisable avant l'exécution du programme.

Si nous interprétons cette erreur de non remplacement du robot en position adéquate pour aborder le motif de l'itération suivante par rapport à la théorie des champs conceptuels, elle correspond à un théorème-en-acte qui est manquant : « Au début de chaque itération, le robot est toujours positionné de la même manière par rapport au motif qu'il va parcourir ». Ce théorème peut être mis en lien avec le concept plus avancé d'invariant de boucle, qui peut être défini simplement comme une propriété qui est vraie avant et après chaque itération.

Nous mettons en relation ce défaut de conceptualisation avec le fait que le programme est perçu comme commandant une suite d'actions du robot virtuel sur la grille, et non comme une succession d'états du système, états liés à l'évolution de variables au cours de l'exécution du programme. La dialectique entre exécution séquentielle d'actions et succession de situations qui présentent des propriétés statiques a été identifiée par Rogalski dans le contexte de l'initiation à la programmation textuelle au lycée (Rogalski, 1987). Dans le contexte de la programmation en langage Scratch d'un robot virtuel sur une grille, la succession d'états est très peu visible pour l'utilisateur. Lorsqu'il lance l'exécution de son programme, celui-ci perçoit une succession de déplacements et d'actions du robot virtuel plutôt qu'une succession d'états de la grille dont le robot, avec sa position et son orientation, fait partie.

Le remplacement du robot par rapport au prochain motif avant chaque itération n'est pas perçu comme une action. Il fait partie du motif algorithmique mais pas du motif visuel. Ce remplacement du robot est donc souvent ignoré, au moins en première intention.

Nous concluons en renseignant les composantes du schème identifié.

#### **Éléments de définition analytique du schème de remplacement du robot relativement au motif**

- But : replacer le robot en position adéquate pour aborder l'itération suivante.
- Règles de conduite de l'action :
  - prise d'information : repérer la position du robot une fois qu'il a parcouru le motif visuel ; puis repérer la position que doit avoir le robot pour aborder le motif suivant ;
  - action proprement dite : à la séquence de blocs déjà présents dans le bloc répéter, ajouter les blocs qui permettent d'amener le robot dans la position voulue ;
  - contrôle : lancer une exécution du début du programme (en mode pas à pas) ou se représenter mentalement cette exécution, afin de vérifier la bonne jonction entre le traitement des deux premiers motifs.
- Invariants opératoires :
  - concept-en-acte : invariant de boucle ;
  - théorèmes-en-acte : « Au début de chaque itération, le robot est toujours positionné de la même manière par rapport au motif qu'il va parcourir ».

## **Conclusion**

Dans cette contribution, nous avons montré comment nous combinons les apports respectifs de la théorie des champs conceptuels et de l'analyse de traces afin d'étudier la résolution de puzzles de programmation avec un langage par blocs. D'un côté, la théorie des champs conceptuels nous donne des clés pour structurer l'analyse de nos données et les interpréter. Réciproquement, la possibilité d'articuler des traitements statistiques sur de larges échantillons avec la précision d'observation d'une approche qualitative est de nature à apporter une robustesse aux résultats établis en se plaçant dans ce cadre d'analyse.

Nous avons mis en œuvre cette méthode sur un exemple de puzzle qui requiert la mobilisation de la notion de boucle. Elle nous a permis d'identifier des erreurs fréquentes d'élèves d'école élémentaire et de collège (7 à 15 ans) lors de la confrontation à cette notion de base de l'algorithmique. À titre d'illustration, nous avons interprété l'une de ces erreurs en termes de schème inopérant et appréhendé la conceptualisation-en-acte sous-jacente à cette difficulté. Une de nos perspectives de recherche est de reproduire cette démarche pour mener des investigations sur l'ensemble des concepts abordés lors de l'initiation à la programmation à l'école obligatoire. Celles-ci nous amènerait à identifier et à documenter un ensemble d'erreurs et de stratégies fréquentes lors de la résolution de puzzles de programmation avec un langage par blocs.

Nous pensons que cette compréhension des erreurs et de la conceptualisation-en-acte des élèves est de nature à outiller les enseignants qui initient leurs élèves à l'informatique. Les résultats obtenus à l'issue de nos recherches pourraient conduire à formuler des recommandations aux enseignants afin de leur permettre de mieux appréhender à la fois les enjeux de savoirs et les difficultés fréquentes pour lesquelles une vigilance particulière s'impose. Par exemple, pour le cas d'étude présenté dans cette contribution, l'erreur est persistante tout au long des années de collège et une intervention pédagogique ciblée serait nécessaire. Un tel document pourrait conseiller aux enseignants de faire remarquer la position initiale du robot, puis de faire exécuter une première fois les instructions présentes dans la boucle avec le mode pas à pas, pour amener à discuter de la position du robot à l'issue de cette exécution par rapport à sa position initiale. La préconisation serait d'en tirer les conséquences en termes de programmation d'une boucle, en attirant particulièrement l'attention sur la jonction entre les deux premières itérations.

Notre travail de recherche vise la compréhension des difficultés inhérentes à l'apprentissage des notions de base en algorithmique dans un environnement de programmation par blocs, et contribue ainsi à étayer les enseignants dans leur compétence à ajuster leurs scénarios pédagogiques et à accompagner leurs élèves pour surmonter ces difficultés.

## Références

- 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.researchgate.net/publication/327302966\\_Computational\\_thinking\\_-\\_a\\_guide\\_for\\_teachers](https://www.researchgate.net/publication/327302966_Computational_thinking_-_a_guide_for_teachers)
- Effenberger, T., et Pelánek, R. (2022). Design and analysis of microworlds and puzzles for block-based programming, *Computer Science Education*, vol. 32, n° 1, p. 66-104. <https://doi.org/10.1080/08993408.2020.1832813>
- Fernandez, C., Freitas, J. A., Lopes, R. d. D., et Blikstein, P. (2022). *Using video analysis and learning analytics to understand programming trajectories in data science activities with Scratch*, IDC '22 : Proceedings of the 21st Annual ACM Interaction Design and Children, Braga, Portugal, p. 253-260. <https://doi.org/10.1145/3501712.3529742>
- Gelman, R., et Gallistel, C. R. (1978). *The child's understanding of number*, Harvard University Press.
- Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., et Rivers, K. (2015). Educational data mining and learning analytics in programming : Literature review and case studies, *Proceedings of the 2015 Innovation and Technology in Computer Science Education (ITiCSE) on Working Group Reports*, p. 41-63. <https://doi.org/10.1145/2858796.2858798>
- Jiang, B., Zhao, W., Zhang, N., et Qiu, F. (2022). Programming trajectories analytics in block-based programming language learning, *Interactive Learning Environments*, vol. 30, n° 1, p. 113-126. <https://doi.org/10.1080/10494820.2019.1643741>
- Khan, M., et Khan, S. S. (2011). Data and information visualization methods, and interactive mechanisms : A survey, *International Journal of Computer Applications*, vol. 34, n° 1, p. 1-14.
- Léonard, M., Peter, Y., Secq, Y., et Fluckiger, C. (2022a). Computational Thinking : Focus on Pattern Identification, *Lecture Notes in Computer Science*, vol. 13 450, p. 187-200. [https://doi.org/10.1007/978-3-031-16290-9\\_14](https://doi.org/10.1007/978-3-031-16290-9_14)
- Léonard, M., Secq, Y., Peter, Y., et Fluckiger, C. (2022b). « Pensée informatique : approche didactique de l'identification de motifs », dans *L'informatique, objets d'enseignement et d'apprentissage. Quelles nouvelles perspectives pour la recherche?*, p. 113-125. <https://hal.science/hal-03697888/>
- Léonard, M., Bouton, M., et Peter, Y. (2023). Détermination de profils relatifs à la mobilisation de schème lors de la résolution de puzzles de programmation, *Actes de la onzième conférence sur*

les environnements informatiques pour l'apprentissage humain, Brest, France.

Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond : an overview of learner modeling techniques, *User Modeling and User-Adapted Interaction*, vol. 27, n° 3, p. 313-350. <https://doi.org/10.1007/s11257-017-9193-2>

Pelánek, R. (2018). Exploring the utility of response times and wrong answers for adaptive learning, *Proceedings of the fifth annual ACM conference on learning at scale*, p. 1-4. <https://doi.org/10.1145/3231644.3231675>

Piaget, J. (1935). *La naissance de l'intelligence chez l'enfant*, Delachaux et Niestlé Neuchatel.

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

Vergnaud, G. (1991). « La théorie des champs conceptuels », dans *Recherches en didactique des mathématiques*, La Pensée Sauvage.

Vergnaud, G. (2001). Forme opératoire et forme prédicative de la connaissance, *Investigações em Ensino de Ciências*, vol. 17, n° 2, p. 287-304.

Vergnaud, G. (2007). Représentation et activité : deux concepts étroitement associés, *Recherches en éducation*, vol. 4, p. 9-22. <https://doi.org/10.4000/ree.3889>

Vergnaud, G. (2013). Qu'est-ce que la pensée?, *La nouvelle revue de l'adaptation et de la scolarisation*, vol. 63, n° 3, p. 277-299. <https://doi.org/10.3917/nras.063.0277>

Vergnaud, G., et Durand, C. (1976). Structures additives et complexité psychogénétique, *Revue française de pédagogie*, p. 28-43. <https://doi.org/10.3406/rfp.1976.1622>

Pour citer ce chapitre :

Léonard, Marielle (2024). « Zoom sur quelques erreurs récurrentes lors des premiers apprentissages en algorithmique », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 135-154. <https://doi.org/10.53480/2024iecare07u/>

## **PARTIE 3**

# **FORMER LES ENSEIGNANTS**



# Enseigner l'informatique à l'école primaire : quelques caractéristiques des représentations des formateurs d'enseignants du premier degré

## 8

Arnauld SÉJOURNÉ<sup>1</sup>  
Emmanuelle VOULGRE<sup>2</sup>  
Jacques BÉZIAT<sup>3</sup>

À la rentrée 2016 entrent en vigueur les nouveaux programmes d'enseignement. Ceux-ci plus encore que les précédents font une place assez large aux contenus et aux compétences numériques, à la fois dans le « socle commun de compétences, de connaissances et de culture » sur les questions d'éducation aux médias et à l'information (EMI), sur les usages des technologies et des ressources numériques en classe dans toutes les disciplines, en termes de compétences à posséder à la sortie de l'école (cadre de référence des compétences numériques – CRCN), et aussi des éléments liés à la discipline informatique elle-même tels que le « codage » et l'initiation à l'algorithmique dès le cours élémentaire, et des activités de robotique (ministère de l'Éducation nationale – MEN, 2016)<sup>1</sup>.

On peut saluer cette consolidation scolaire des connaissances et des compétences liées au domaine informatique et aux technologies numériques, mais la formation des enseignants, initiale et continue, permet-elle la mise en œuvre dans les classes de telles ambitions ? La question n'est pas nouvelle. Le certificat informatique et Internet (C2i) des années 2000 puis l'actuel Pix sont sans doute une partie de la réponse pour l'attestation de compétences acquises en formation ou par soi-même. Pour autant, force est de constater la timidité des usages numériques dans les classes à l'école primaire, sur le terrain, et la relative application des programmes scolaires quant aux contenus liés aux technologies numériques (approches intégratives et médiatiques) et, sans doute davantage pour les contenus spécifiquement liés au domaine informatique.

Notre étude se propose de mettre ce point en lumière à travers ce qu'en disent les formateurs d'enseignants du premier degré, notamment les enseignants référents pour les usages du numérique (ERUN) et les conseillers pédagogiques premier degré (CP1D), et en tentant d'appréhender ce qu'ils comprennent eux-mêmes de ce qu'est l'informatique à l'école.

1. Université de Nantes, EA 2661  
CREN - Centre de Recherche en  
Éducation de Nantes, 44000 Nantes,  
France

2. Université Paris Cité, Laboratoire  
EDA, 75006 Paris, France

3. Université de Caen, EA 7454  
CIRNEF - Centre Interdisciplinaire  
de Recherche Normand en Éducation  
et Formation, 14000 Caen, France

1. <https://eduscol.education.fr/document/15409/download>



## Cadre général pour les technologies numériques et l'informatique à l'école

### Pensée informatique, penser l'informatique à l'école

Depuis les prémices en 1970 de l'introduction de l'informatique à l'école primaire, les mentalités et les discours sociaux évoluent. La nécessité de former les élèves aux outils du monde moderne et les initier à une certaine culture technique semble faire consensus. Nous nous arrêtons ici sur deux rapports, l'un émanant de la société civile (CESE) et l'autre de l'Inspection générale.

2. CESE : Conseil Économique Social et Environnemental <https://www.lecese.fr/>. Et pour les questions liées aux technologies numériques à l'école : <https://www.lecese.fr/travaux-publies/lecole-lere-du-numerique>

En 2021, l'avis du Conseil économique, social et environnemental CESE<sup>2</sup> présenté par Marie-Pierre Gariel au nom de la section de l'éducation, de la culture et de la communication « L'école à l'ère du numérique » propose 20 recommandations à l'État français. Deux d'entre elles nous semblent intéressantes à prendre en compte pour notre étude sur la formation des enseignants à l'informatique. Concernant l'axe B « Mettre en place une gouvernance partagée et soutenir la souveraineté numérique dans le domaine éducatif », la CESE préconise que « l'État et les collectivités territoriales engagent véritablement des politiques publiques de co-construction dans le domaine du numérique éducatif s'appuyant sur des orientations nationales, claires et ambitieuses, compatibles avec les objectifs de développement durable, en matière d'équipements et contenus numériques, d'usages du numérique éducatif, de formation et de pédagogie avec le numérique (préconisation B.3, p. 6). Concernant l'axe C « Renforcer la formation et l'accompagnement des enseignantes, des enseignants et des familles », la CESE préconise qu'« en fonction des besoins individuels et de ceux des équipes pédagogiques, de développer la formation continue des enseignantes et des enseignants aux spécificités pédagogiques et didactiques de l'enseignement avec le numérique pour qu'ils puissent faire évoluer de façon raisonnée leurs pratiques pédagogiques en présentiel mais aussi en distanciel » (C.8, p. 7). Pour autant, ce rapport ne définit pas ce qu'est l'enseignement de l'informatique, et encore moins à l'école primaire.

3. IGESR Inspection générale de l'éducation, du sport et de la recherche intitulé « Pratique de l'informatique aux cycles 3 et 4 » (N° 21-22 169A) PDF, 77 p. novembre 2022. <https://www.education.gouv.fr/media/120388/download>

Le second rapport porte spécifiquement sur l'enseignement de l'informatique à l'école primaire et formule aussi des recommandations. Il s'agit d'un rapport de l'Inspection générale de l'éducation, du sport et de la recherche intitulé « Pratique de l'informatique aux cycles 3 et 4 » (N° 21-22 169A novembre 2022)<sup>3</sup>. Les auteurs préconisent de distinguer les apprentissages et les compétences à cibler selon les objectifs d'enseignement relatifs à deux champs, celui de la pensée informatique et celui de la littératie numérique. De plus, il est en particulier recommandé de former les conseillers pédagogiques pour qu'ils puissent accompagner les enseignants pour l'éducation à la pensée informatique en classe (p. 9). Ce faisant, ce

rapport met en perspective les usages de l'informatique observés en cycle 3 (robotique ou dans les disciplines) avec les enjeux liés au développement d'une pensée informatique chez les élèves, pour constater l'insuffisance de la formation des enseignants dans le domaine.

Ce rapport donne deux définitions de la pensée informatique, l'une didactique et l'autre institutionnelle. Dans la première, il s'agit d'un « processus de pensée mobilisant abstraction et décomposition. Il s'agit d'une approche théorique dont l'opérationnalisation ne va pas de soi ». Ainsi « la pensée informatique peut être présentée comme un processus intellectuel visant à englober la définition, la formulation des problèmes, leur résolution mais aussi l'expression de solutions de manière à ce qu'ils puissent à la fois être pris en charge par un dispositif de traitement automatisé (ordinateurs, applications logicielles) et être exécutés ». Pour la seconde, la pensée informatique est ici « considérée utile pour faire face à la réalité actuelle et à venir, comme une compétence cruciale pour les citoyens et professionnels du XXI<sup>e</sup> siècle ». Il est aussi introduit l'idée qu'un « continuum de formation s'est développé et structuré ». Nous ne sommes pas certains que ce continuum soit effectif dans les curricula de formation des enseignants. Ces curricula, ou leur application, tombent parfois dans les travers dénoncés dans ces deux rapports, travers liés à des usages de technologies informatiques utilitaristes ou ludiques, déconnectés de leur discipline berceau : l'informatique.

## **Quelques éléments pour penser l'informatique**

Cette discipline est donc à penser en tant que telle. En effet « l'informatique est un domaine qui est souvent confondu avec ses applications, notamment car tout le monde utilise ses supports techniques (ordinateur, tablette, téléphone) et ses applications (éditeurs de texte, tableaux intégrant des formules de calculs, navigateurs Internet, logiciels de retouche photo, etc.) » (Tchounikine, 2016, p. 5). En réalité, l'informatique est liée à la résolution de problèmes. Dowek (2011, p. 21) propose de la définir en quatre concepts : l'algorithmique, les machines, le langage et l'information. L'algorithme permet de résoudre certains problèmes de manière systématique. Les langages permettent de communiquer ces algorithmes aux machines en charge de les exécuter. Ces machines, paramétrables, sont des « systèmes physiques, pour lesquels un protocole d'interaction permet d'échanger des données » tels que les ordinateurs (p. 23). Le langage et l'information (ici les données traitées et calculées) sont liées « car ce sont deux instances d'une même idée : exprimer symboliquement des algorithmes, pour l'une ; des données, pour l'autre » (p. 28).

Éric Bruillard (2010) propose d'élargir la focale. Pour lui, se plaçant du point de vue des concepteurs/utilisateurs et de leurs activités, l'informatique ne se réduit pas à la science informatique. Il propose donc de caractériser l'informatique à partir de trois approches imbriquées (p. 5-7) : (1) l'informatique comme une science du calcul, faisant le lien entre calculs et raisonnements, « une science des algorithmes » ; (2) l'informatique qui produit des résultats pour faciliter l'activité humaine, et qui engage sa perception dans un cycle action humaine-traitement-présentation d'un état côté machine ; et (3) « une informatique plus relationnelle, en interaction large avec l'environnement, communication et traitement étant étroitement associés. Cela inclut des formes collectives de traitement, les traitements humains, individuels et collectifs, s'interpénétrant avec des traitements automatiques, rendant parfois difficile de démêler les uns des autres. C'est le domaine de l'informatique "sociale" ».

Ce faisant, Bruillard (2010) invite à sortir d'une opposition stérile entre une informatique « vue comme une science » ou « comme un ensemble de technologies ». Il propose de « se focaliser sur le traitement de l'information par les humains utilisant les ordinateurs ou plus largement les objets informatisés » (p. 7). Cette approche n'est pas sans conséquences d'un point de vue didactique, ni en termes de formation des enseignants.

### **Enseigner l'informatique à l'école primaire, et y former**

L'histoire de l'informatique à l'école, largement relatée par ailleurs, a été marquée par les flux et les reflux d'intérêt de la part des pouvoirs publics et par les discours sociaux depuis les années 1980, qui ont apporté leur lot de confusions sur ce qu'est l'informatique et sur ce qu'il faut en faire et en enseigner à l'école (Béziat, 2012). Nous n'en commenterons pas les débats ici. Retenons simplement que pour qu'une didactique de l'informatique (et donc d'une formation des enseignants à cette didactique) existe, il faut qu'il y ait consensus social sur la nécessité de former les jeunes aux mondes numériques (Fluckiger, 2019). Le creuset de cette didactique est à la croisée des technologies et des sciences humaines, et donc de différentes cultures, différentes approches des réalités sociales et techniques (Cardon, 2019).

4. Pour l'auteur (p. 3), l'agentivité dans un environnement numérique s'exprimerait en ces termes : « s'exprimer dans un monde numérique, s'informer, garder le contrôle de son image, maîtriser la diffusion de ses données personnelles, mais aussi savoir les exploiter. Elle concerne aussi la capacité à créer et à produire du numérique et avec le numérique, à résoudre des problèmes ».

Dans le même sens, Drot-Delange et Fluckiger (2022) définissent l'informatique comme « l'ensemble des savoirs et savoir-faire nécessaires à chacun pour qu'il puisse être en mesure d'exprimer son agentivité<sup>4</sup> dans la société numérique actuelle » (p. 2). Il s'agirait de « penser avec l'informatique ». De manière extensive, cette « culture de l'informatique » peut être pensée en trois éléments (Bruillard, 2014, p. 3) : pensée informatique, maîtrise des objets

informatiques, et participation à des activités sociales dans un monde en réseau.

Enseigner l'informatique et les technologies numériques amène sans doute à prendre en compte ces trois polarisations et leurs articulations. Cela dit, et c'est le sens de cette étude, tenant compte de cette approche, les enseignants et leurs formateurs sont-ils suffisamment formés ? Les travaux instruisant la question laissent penser que non. Les enseignants qui intègrent des situations d'enseignement de l'informatique dans leur classe bricolent, à partir du matériel disponible et de ce qu'ils comprennent de l'informatique (Béziat, 2019). Les formations elles-mêmes portent souvent sur une conception utilitariste des appareils informatiques et une approche disciplinaire des ressources numériques, parfois en lien avec l'éducation aux médias et à la citoyenneté, mais plus rarement au service du développement d'une culture technique intégrant les différents aspects liés aux environnements numériques et à l'informatique. Les enseignants sont formés aux technologies numériques à partir de ce que leurs formateurs comprennent de l'informatique. Ce sont ces formateurs, les ERUN et les conseillers pédagogiques du premier degré, que nous interrogeons ici.

## Méthodologie

Notre démarche de recherche est exploratoire et vise à décrire et analyser la manière dont un groupe de professionnels, les formateurs en technologies de l'information et de la communication pour l'éducation (TICE) du premier degré, issus de deux territoires académiques, se représentent l'enseignement de l'informatique à l'école primaire. Nous présenterons successivement dans cette partie, le public cible, la méthodologie de recueil de données ainsi que la méthodologie d'analyse.

### Public cible : les formateurs TICE

Le groupe de professionnels interrogé est composé d'ERUN et CPID chargés de mission numérique.

Les ERUN<sup>5</sup> sont des enseignants du premier degré qui exercent des missions de formateurs, déchargés de tout ou partie de leur service d'enseignement. Ils sont membres d'une équipe de circonscription<sup>6</sup> et travaillent sous l'autorité conjointe de l'inspecteur académique – directeur académique des services de l'Éducation nationale (IA-DASEN) et de l'inspecteur de l'Éducation nationale (IEN) de circonscription. La thèse de Villemonteix (2007) dénombre 34 appellations de l'« animateur informatique » à

5. ERUN : des enseignants référents pour les usages du numérique à l'école, <https://www.tice-education.fr/tous-les-articles-er-res-sources/articles-informatiques/1648-erun-des-enseignants-referents-pour-les-usages-du-numerique-a-lecole>

6. Une circonscription est un découpage arbitraire du territoire départemental en territoires académiques plus petits.

7. Tableau des missions de l'ERUN de la DSDEN de la Sarthe, [https://www2.dsden72.ac-nantes.fr/medias/fichier/enseignement-mission-erun\\_1648731851320-pdf?ID\\_FICHE=659129&INLINE=FALSE](https://www2.dsden72.ac-nantes.fr/medias/fichier/enseignement-mission-erun_1648731851320-pdf?ID_FICHE=659129&INLINE=FALSE)

8.

<https://www.education.gouv.fr/bo/15/Hebdo30/MENE1516648C.htm>

l'« animateur TICE » (ATICE) ou encore « référent TICE ». Les missions professionnelles se sont précisées au fil du temps et semblent davantage s'orienter vers les usages pédagogiques des technologies. Par exemple, pour le département de la Sarthe, la mission de l'ERUN est l'accompagnement et le suivi des enseignants dans le développement des usages du numérique dans les pratiques d'apprentissage et d'enseignement ; l'analyse et la production de ressources pédagogiques. Parmi les principales compétences et expériences, il est attendu par exemple une expertise en matière d'usage du numérique avec les élèves ; une attitude de veille pédagogique et institutionnelle ; des capacités à produire, partager et mutualiser des documents, informations et ressources dans le cadre d'un travail en réseau avec l'utilisation des outils de travail collaboratif<sup>7</sup>. Il est à noter que parmi les missions proposées aucune ne porte explicitement sur l'informatique (en tant que science) à l'école primaire.

Les missions<sup>8</sup> des CPID sont décrites dans la circulaire du 21 juillet 2015. Le CPID est totalement déchargé de service d'enseignement en classe et travaille à temps plein au service de la mise en œuvre de la politique académique. Concernant l'accompagnement pédagogique des maîtres et des équipes, le CPID « assure l'accompagnement professionnel des maîtres et des équipes pédagogiques dans la mise en œuvre des programmes d'enseignement et des projets de classe, de cycle ou d'école ». Il est un formateur polyvalent, expert et expérimenté qui « contribue à la mise en œuvre et à l'accompagnement des évolutions de la politique éducative ».

## Corpus recueilli

Nous avons élaboré un questionnaire composé initialement de 43 questions ouvertes et fermées portant sur différentes thématiques. Pour cette étude, nous nous sommes focalisés sur les 14 premières questions portant sur les thématiques suivantes.

## Méthodologie d'analyse

Nous inscrivons notre analyse à la suite des travaux portant sur les représentations professionnelles. Nous intéressant à un groupe de professionnels inscrits dans des contextes variés concernant un objet professionnel particulier, former à l'enseignement de l'informatique à l'école primaire, nous nous focalisons sur leurs représentations professionnelles. Ces dernières « ne sont alors plus un savoir de sens commun [comme pourraient l'être les représentations sociales, note des auteur(e)s] mais possèdent un rapport de sens et d'implication dans une activité professionnelle.

Les représentations professionnelles sont fortement contextualisées, elles se construisent et s'utilisent au cours d'échanges entre pairs et uniquement dans ce contexte, elles sont spécifiques au contexte professionnel où elles prennent sens. De par ce fait, elles sont prescriptives des conduites » (Blin, 1997). Ces représentations sont alors constitutives du savoir professionnel, ici des formateurs, qu'ils mettent en œuvre pour traiter des situations de travail (Lefeuvre *et al.*, 2009). Il s'agit ici, en identifiant les représentations des formateurs, d'explorer leurs intentions en amont de leurs actions et qui leurs donnent de la cohérence (Robert et Robinet, 1992). Les représentations des formateurs portant sur « enseigner l'informatique à l'école primaire » ainsi que les différentes activités proposées ont été catégorisées (Bardin, 2013) à partir des définitions données en particulier par Dowek (2011), Bruillard (2014) et Drot-Delange (2016) introduites dans la partie 2.

## Représentations des formateurs à propos de l'informatique à l'école

### Formateurs et leurs profils

Pour les CP1D, six sur un sont en poste depuis moins de quatre ans, deux entre cinq et neuf ans, six ont plus de dix ans d'expérience. Ils participent à différents groupes de travail dont 11 sur 14 au groupe « valorisation des usages numériques », pour la moitié au groupe regroupant les thématiques « programmation, robotique, objets connectés » spécifique à l'académie de Versailles, un seul au groupe EMI de l'académie de Nantes.

Concernant les ERUN, trois sur huit ont un mi-temps en classe et un mi-temps pour différentes missions au service de l'IEN de circonscription et du CP1D, dont celle de formation. Cinq sur huit sont formateurs depuis moins de deux ans. Ils participent aussi à différents groupes de travail. Trois sur huit contribuent au groupe « programmation, robotique, objets connectés » ; tous les huit participent au groupe « valorisation des usages numériques » et trois sur les huit participent aussi groupe EMI.

Qu'il s'agisse des CP1D ou des ERUN, ils déclarent majoritairement se former seuls, au moyen de tutoriels, par les échanges avec les pairs, en montant des projets pour la classe ou en aidant les collègues. Un seul a suivi une formation universitaire en informatique. Deux formateurs précisent qu'ils ont suivi des formations (intitulées « plan numérique », « enseignants innovants »).

De plus, ils ne se perçoivent pas en tant que « formateur pour l'enseignement de l'informatique » ; cet enseignement n'étant,

selon eux, pas dans les programmes de l'école primaire. Certains n'enseignent pas l'informatique en tant que telle (CP1D EMI).

Alors, comment se représentent-ils ce que pourrait être enseigner l'informatique à l'école ?

## **Formateurs et leurs représentations de l'informatique à l'école primaire**

Les formateurs formulent plusieurs types de réponses à propos de l'enseignement de l'informatique. Nous avons distingué des indicateurs thématiques en lien avec les définitions données précédemment : relatifs aux outils d'enseignement, à leur maîtrise, à la connaissance technique sur les outils, aux connaissances ou notions en informatique, puis ceux concernant des finalités disciplinaires, enfin des modalités pédagogiques.

### **Enseigner et apprendre au moyen d'outils (matériels et logiciels)**

Un seul formateur (CP1D), participant au groupe « programmation, robotique, objets connectés », évoque l'enseignement de l'informatique en référence aux outils pour l'enseignement : « découverte du poste informatique, utilisation de logiciels de bureautique, de modélisation 3D ». Dans une première analyse, « découverte du poste informatique et utilisation de logiciels de bureautique » renvoie à ce que peut dire Tchounikine (2016) ; l'informatique est souvent confondue avec le matériel et ses applications. Toutefois, cela aurait aussi pu s'entendre comme la connaissance de l'objet technique (fonction et fonctionnement) en référence aux attendus des programmes de l'école primaire (cycle 3, sciences et technologie).

### **« Maitriser » des outils et développer de compétences techniques**

Deux CP1D considèrent que l'enseignement de l'informatique à l'école concerne « la maîtrise des outils » et le « développement de compétences techniques », dont celles de savoir utiliser les logiciels usuels et de se représenter leur fonctionnement. Par exemple, l'un des formateurs déclare « savoir recopier un document en traitement de texte, nommer et ranger des dossiers », « savoir envoyer un mail... utiliser un traitement de texte » (CP1D 13), « fonctionnement des objets connectés » (CP1D 7).

Nous retrouvons ces représentations chez cinq ERUN. « L'enseignement de l'informatique est plutôt consacré à l'étude des

outils (matériel), [...] : "comment fonctionne, comment faire..." » (ERUN 60), « L'enseignement informatique est centré sur les outils » (ERUN 71), « Support pour l'information » (ERUN 25), « L'enseignement de l'informatique est plutôt consacré à l'étude des outils » (ERUN 60). Ils évoquent des situations tels qu'« effectuer des recherches sur Internet et créer des documents » (ERUN 28), « utiliser le tableur en mathématique » (ERUN 60, ERUN 71), « créer un document en utilisant des compétences de gestion de fichiers (enregistrer, copier/coller, mettre en forme, insérer) » (ERUN 28), « créer un tableau, supprimer des colonnes, des lignes, utiliser les fonctions, trier des données » (ERUN 19).

### **Enseigner des notions en informatique explicitées dans les programmes**

Pour quatre CPID, l'enseignement de l'informatique concerne l'enseignement de notions en informatique comme pour le CPID 7 qui déclare « Enseigner l'informatique, pour moi, désigne l'enseignement de la programmation, de l'algorithmique, de la robotique, et donc du langage, mais aussi des "machines" qui utilisent ces langages » puis (CPID 14) « codage et programmation uniquement ». Ils évoquent différentes situations d'enseignement telles que « Séquence de programmation : [...] séances de déplacements (cycles 1 et 2) » (CPID 16), « l'enseignement d'un langage de programmation spécifique » (CPID 12).

Nous retrouvons les mêmes représentations chez quatre ERUN. Par exemple, « l'enseignement à l'informatique concerne des notions liées à la programmation [...] [à] Internet » (ERUN 28); « L'enseignement de l'informatique est plutôt consacré à l'apprentissage de la programmation » (ERUN 60). Les situations d'enseignement citées sont par exemple « la robotique qui donne à étudier la pensée informatique et sa logique » (ERUN 27), « Enseigner le code au cycle 3, se lancer dans la programmation » (ERUN 27), « programmer avec ScratchJr et Scratch des histoires étudiées en classe » (ERUN 28).

### **Finalités disciplinaires ne concernant pas le primaire**

Enfin, pour trois CPID et un ERUN, l'enseignement de l'informatique concerne des finalités disciplinaires mais pas au primaire. « L'informatique est plus spécifique, c'est un enseignement disciplinaire à part entière » (CPID 15, CPID 19), « l'informatique [n'est pas un] des objets d'enseignement à proprement parler dans le 1<sup>er</sup> degré » (CPID 35), « l'informatique est une matière en soi » (ERUN 27).



### À propos des modalités pédagogiques

Pour quatre ERUN, l'enseignement de l'informatique permet d'évoquer des modalités pédagogiques tels que des démarches de type concours, défi, mais aussi des démarches pluridisciplinaires : « Programmer à l'aide de Scratch ou ScratchJr des histoires étudiées en classe » (ERUN 28), « Utilisation des ordinateurs fond de classe » (ERUN 37), « Concours ScratchJr » (ERUN 21), « Participer à un défi de programmation à l'école élémentaire » (ERUN 71). Comme le soulignent différents formateurs, cette dynamique de concours ou de défis, au-delà de la motivation de l'élève, contribue pour l'enseignant à « fédérer la classe autour d'un projet pluridisciplinaire », à « valoriser le travail des enseignants et des élèves » et donne au formateur « la satisfaction de la participation des classes et le suivi des différentes épreuves et des questions qui arrivaient. »

Notons qu'aucun élément, du domaine 3 : « la formation de la personne et du citoyen », n'est mentionné.

### Formation de formateurs axée sur les usages du numérique

Concernant des formations qu'ils allaient ou avaient mises en œuvre durant l'année 2020 (cf. tableau 1), sept formateurs ont décrit leurs formations (20 recueillies). Parmi les réponses, nous avons identifié une place réduite des formations concernant en particulier l'initiation à l'algorithmique et la programmation comparée à celle portant sur les usages du numérique, l'EMI, l'initiation à des outils (Ipad, espace numérique de travail – ENT, vidéo projecteur interactif – VPI, tableau numérique interactif – TNI).

TABLEAU 1 – Formation mise en œuvre.

Thématiques des formations	Nombre de formations	Exemple d'intitulé de la formation	Objectifs de la formation
Initiation algorithmique et programmation	2	Formation à la programmation ; Enseigner le code au cycle 3	Amener les enseignants à s'approprier le projet de rencontre robotique, se lancer dans la programmation
EMI	5	Les médias scolaires ; Trouver l'information	Construire un média scolaire, savoir trouver une information fiable sur le net, la trier la vérifier
Formation « les outils au service des disciplines »	9	Formation aux usages du TNI	Découvrir des usages pédagogiques du TNI
Formation aux outils	4	Le carnet de suivi des progrès Ipad	Construire et transmettre un carnet de suivi au format numérique Ipad

## Besoins de formation rarement exprimés dans le domaine de l'informatique

Nous avons interrogé les formateurs sur les principales compétences qu'ils souhaiteraient développer. Treize ont répondu à cette question. Deux personnes précisent qu'elles souhaiteraient développer leurs compétences dans le champ de l'informatique comme par exemple « complément de formation autour de la programmation et du codage ».

Pour deux autres, il s'agit de formation sur les outils que sont l'ENT, la webradio et la vidéo.

Pour près de la moitié des collègues (6 sur les 13), les compétences à développer relèveraient principalement des quatre domaines de compétence du formateur<sup>9</sup> : (1) penser – concevoir et élaborer ; (2) mettre en œuvre – animer ; (3) accompagner l'individu et le collectif ; (4) observer – analyser – évaluer. Nous donnons ci-dessous quelques exemples.

9. Mission encadrée par différents textes dont celui du 23 juillet 2015.

- « Former à distance (Magistère...) Connaître différents modes de formation Évaluer l'impact de ma formation » ;
- « J'ai besoin de travailler encore la communication et la pédagogie active de la formation. » ;
- « Techniques de mise en situation de formation » ;
- « Favoriser encore davantage les interactions entre apprenants ».

Enfin, pour deux formateurs, il s'agirait de développer des compétences en ingénierie de la formation.

## Discussion, conclusion et perspectives

Cette étude se proposait de comprendre comment des formateurs de la formation continue des enseignants du primaire se représentent l'informatique afin de nourrir leur réflexion sur leurs actions au service de la politique académique de formation et d'accompagnement des enseignants en formation continue. Nos résultats montrent notamment que globalement, les représentations professionnelles de l'informatique à l'école des ERUN et des CPID se recouvrent et dialoguent principalement avec les attendus institutionnels, et encore pas suffisamment avec la manière dont la didactique de l'informatique définit l'informatique scolaire.

L'analyse des énoncés recueillis concernant les représentations professionnelles des formateurs, CPID et ERUN confondus, mettent en lumière différentes dimensions de l'« informatique » scolaire vue précédemment : (1) en tant qu'outils pour enseigner et apprendre tout en les appréhendant (fonction et fonctionnement) ;

10. <https://eduscol.education.fr/document/52689/download?attachment>

(2) en tant que compétences dans la « maîtrise » de ces outils (fonctionnement) lesquelles seraient au service des disciplines ; (3) en lien avec la résolution de problème, par la même impliquant une démarche de construction d'une pensée informatique, lors d'activités branchées (Scratch), d'activités robotiques et débranchées sous des modalités pédagogiques diverses telles que les défis, projets pluridisciplinaires, concours. Nous retrouvons dans leurs réponses les attendus institutionnels<sup>10</sup>. Dans le programme du cycle 3 (CM1, CM2 et 6<sup>e</sup>), l'informatique n'est pas une discipline en soi mais est abordée comme une notion transversale en tant que « contribution essentielle » pour que les élèves puissent « comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques » (p. 7). C'est ce qu'en particulier les formateurs du groupe « programmation, robotique, objets connectés » énoncent à juste titre en précisant que l'informatique n'est pas une discipline enseignée à l'école primaire. Toutefois, pour préparer les professeurs des écoles à questionner l'environnement numérique dans lequel les élèves de l'école primaire évoluent, communiquent, partagent, créent, dans quelles mesures la formation de ces derniers, déjà en poste, ne devrait-elle pas donner lieu à développer des compétences tant en informatique que sur les usages sociaux du numérique (Bruillard, 2014) tout en les articulant ? De plus, les formateurs des deux académies sont inscrits dans différents groupes thématiques dont pour moitié celui de la « valorisation des usages du numérique » et celui intitulé « programmation, robotique et objets connectés ». Ce faisant, est-ce à dire qu'ils construisent des ponts au sein des séquences de formation entre ce qui porte sur l'apprentissage de l'informatique et ce qui se réfère aux usages des technologies de l'information et de la communication ? Néanmoins, au regard des formations prévues ou données, cette articulation ne paraît pas visible ni au niveau des titres des formations proposées, ni des objectifs exprimés. Nous percevons bien ici la limite du questionnaire pour aller plus en profondeur dans notre compréhension des représentations professionnelles de l'informatique par les formateurs.

Ensuite, concernant les formations mises en œuvre par les ERUN et les CP1D, nous avons identifié que celles-ci portaient en particulier sur « des savoir-faire techniques » permettant d'utiliser les technologies plutôt institutionnelles (ENT, TNI), pour la valorisation des usages du numérique dans différentes disciplines et plus rarement au service de l'explicitation de la pensée informatique. Au regard des missions des ERUN en tant qu'« enseignants référents pour les usages numériques » et des CP1D qui « assurent l'accompagnement professionnel des maîtres et des équipes pédagogiques dans la mise en œuvre des programmes d'enseignement et des projets de classes », cela nous semble tout à fait en adéquation. Ainsi, tout en tenant compte du faible nombre de réponses des formateurs, ce résultat nous laisserait penser que former les enseignants à

L'informatique à l'école ne serait pas plus une priorité que de former aux usages du numérique. Au regard des formateurs questionnés et des académies d'origine, ceci pourrait résulter des choix politiques des territoires académiques, par exemple l'académie de Versailles a mis en place un groupe de formateurs sur la thématique « programmation, robotique et objets connectés » (groupe ROC), des ressources humaines disponibles sur le territoire, des dispositifs prévus de formation des formateurs.

Enfin, nous avons vu que pour certains formateurs, le développement de compétences dans le champ de l'informatique ne semble pas une priorité, et qu'ils répondaient aux attentes institutionnelles, notamment au niveau des politiques éducatives territoriales, aussi en fonction des besoins identifiés sur le terrain, en matière de contenus et de démarches de formation hybride. Ainsi, si cette étude donne l'occasion de porter un regard sur le champ de la formation continue des enseignants, en particulier sur la dimension de l'informatique à l'école primaire, elle interroge plus largement son adéquation aux besoins présents et futurs des citoyens évoluant au sein d'une société « numérique ».

La posture des ERUN et CP1D semble donc répondre aux besoins immédiats du terrain et être le relais des orientations politiques et institutionnelles. C'est à la fois cohérent et pragmatique, mais peut-on penser que cela suffise au développement de pratiques de classes à la fois qui se généralisent et qui répondent aux besoins d'éducation des élèves aux enjeux des environnements informatisés dans lesquels ils sont appelés à agir ? Ces compétences du XXI<sup>e</sup> siècle (Tremblay et Poellhuber, 2022), en partie préfigurées dans le « socle commun de connaissances, de compétences et de culture » et dans le « cadre de référence des compétences numériques (CRCN) », peuvent-elles être pleinement développées quand les formateurs, eux-mêmes insuffisamment formés, accompagnent les enseignants sur leurs besoins immédiats, liés aux contraintes de classe et au matériel disponible ? Pour appréhender le périmètre et les enjeux d'une éducation, et donc de la formation des enseignants, aux technologies numériques et à l'informatique dans leurs aspects disciplinaires, instrumentaux, sociaux, sociétaux, éthiques et citoyens, sans doute faut-il s'extraire suffisamment des contingences pratiques immédiates. Une éducation au « numérique » en classe passe nécessairement par des enseignants formés, dotés et accompagnés et des formateurs, eux-mêmes formés, et qui puissent à la fois s'approprier le terrain tout en appréhendant la diversité des situations pédagogiques possibles liées aux technologies numériques et à l'informatique, en cohérence avec les évolutions technologiques de ce monde, et de manière prospective.

## Recommandations

Au regard des recommandations des représentants de la société civile, de l'institution scolaire, de la recherche, des transformations rapides de l'environnement numérique dans lequel les citoyens évoluent, il est urgent que l'État s'engage à la mise en place de politiques publiques ambitieuses, définies collectivement et durables pour le contexte de l'école primaire (le rapport sur l'enseignement de l'informatique en France par l'Académie des sciences en 2013 titrait : « Il est urgent de ne plus attendre »).

La priorité première devrait être donnée à la formation initiale et continue des enseignants du primaire à l'enseignement de l'informatique. Pour cela, il nous semble nécessaire, de penser son « inclusion » non en tant que nouvelle discipline telle que cela a pu être proposé dans le rapport de l'Académie des sciences (2013) mais de travailler la mise en dialogue des « cultures numériques » avec les différents champs disciplinaires enseignés, cela dans une perspective des humanités numériques (Allouche, 2021 ; Séjourné *et al.*, 2023). En accord avec Baron (2021), ces « cultures numériques » sont composées d'au moins trois dimensions : « citoyenne » en référence aux usages numériques, « informatique » concernant les concepts en informatique, « technique » qui renvoie à l'objet technique (fonction et fonctionnement).

## Références

Académie des sciences (2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*, Rapport de l'Académie des sciences. [https://www.academie-sciences.fr/pdf/rapport/rads\\_0513.pdf](https://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf)

Allouche, E. (2021). Les humanités numériques, pour un dialogue interdisciplinaire entre recherche et éducation, *Frantice.net*, n° 17. <https://hal.science/hal-03080381>

Bardin, L. (2013). *L'analyse de contenu*, Presses universitaires de France. <https://doi.org/10.3917/puf.bard.2013.01>

Baron, G-L. (2021). Brèves réflexions sur les humanités numériques, *Frantice.net*, n° 17. <https://hal.science/hal-03080381>

Béziat, J. (2012). Informatique, outil ou objet ? Permanence d'une question. Le cas de l'école primaire en France, *Adjectif : analyses et recherches sur les TICE*, vol. 3. <http://www.adjectif.net/spip/spip.php?article177>

Béziat, J. (2019). À l'école primaire, robotique éducative en milieu ordinaire, *Spirale - revue de recherche en éducation*, vol. 63. <https://doi.org/10.3917/spir.063.0091>

Blin, J.-F. (1997). *Représentations, pratiques et identités professionnelles*, L'Harmattan.

Bruillard, E. (2010). « Acteurs et territoires de l'éducation à l'information : un point de vue 'informatique' », dans Françoise

Chapron et Éric Delamotte (dir.), *L'éducation à la culture informationnelle*, Presses de l'Enssib, p. 68-75. <https://doi.org/10.4000/books.presseenssib.837>

Bruillard, E. (2014). *Une voie pour penser et construire une formation à l'informatique pour les élèves de l'école primaire ?* <https://hal.science/hal-03948939v1>

Cardon, D. (2019). « Introduction. Coder, décoder », dans : Dominique Cardon (dir.), *Culture numérique*, Presses de Sciences Po, p. 5-13. <https://www-cairn-info.ezproxy.u-paris.fr/culture-numerique-9782724623659-page-5.htm>

Dowek, G. (2011). Les quatre concepts de l'informatique, *Sciences et technologies de l'information et de la communication en milieu éducatif*, EduTice, p. 21.29. <https://edutice.archives-ouvertes.fr/edutice-00676169>

Drot-Delange, B., et Fluckiger, C. (2022). « Dialogue sur la didactique de l'informatique : Bilan, enjeux et perspectives », dans Éric Delamotte (dir.), *Recherches francophones sur les éducations aux médias, à l'information et au numérique : points de vue et dialogues*, Presses de l'Enssib, p. 216-235. <https://doi.org/10.4000/books.presseenssib.17229>

Fluckiger, C. (2019a). *Une approche didactique de l'informatique scolaire*, Presses universitaires de Rennes.

Lefevre, G., Garcia A. et Namolovan, L. (2009). Les indicateurs de développement professionnel, *Questions Vives*, vol. 5, n° 11. <https://doi.org/10.4000/questionsvives.627>

Légifrance (2013). *Loi d'orientation et de programmation pour la refondation de l'école de la république (8 juillet 2013)*. <https://www.legifrance.gouv.fr/dossierlegislatif/JORFDOLE000026973437/>

MEN (2016). *Initiation à la programmation, aux cycles 2 et 3*, Éduscol. <https://eduscol.education.fr/document/15409/download>

MEN (2020). *Programmes d'enseignement*, Éduscol. <https://eduscol.education.fr/74/j-enseigne>

Robert A. et Robinet J. (1992). Représentations des enseignants et des élèves, *Repères Irem*, vol. 7, p. 93-99.

Séjourné, A., Chen, Y., Voulgre, E., Brilland, X., et Brothier, S. (2023). Formation MEEF LVE-TICE et Humanités numériques, *Productions des groupes thématiques numériques (Direction du numérique pour l'éducation du Ministère de l'Éducation nationale et de la Jeunesse)*. <https://hal.science/hal-03938949v1>

Tchounikine, P. (2016). *Initier les élèves à la pensée informatique et à la programmation avec Scratch*. <https://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.pdf>

Tremblay, C. et Poellhuber, B. (2022). Analyse qualitative de référentiels de compétences du XXI<sup>e</sup> siècle, numériques et informationnelles : tendances mondiales observées, *Formation et profession*, vol. 30, n° 2, p. 1-26. <https://doi.org/10.18162/fp.2022.648>

Villemonteix, F. (2007). *Les animateurs TICE à l'école primaire : spécificités et devenir d'un groupe professionnel : analyse de processus de professionnalisation dans une communauté de pratiques en ligne*, thèse de l'université René Descartes – Paris V. [https://theses.hal.science/file/index/docid/202443/filename/These\\_Francois\\_Villemonteix.pdf](https://theses.hal.science/file/index/docid/202443/filename/These_Francois_Villemonteix.pdf)

Pour citer ce chapitre :

Séjourné, Arnauld, Voulgre, Emmanuelle, et Béziat, Jacques (2024). « Enseigner l'informatique à l'école primaire : quelques caractéristiques des représentations des formateurs d'enseignants du premier degré », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 157-172. <https://doi.org/10.53480/2024iecare08v>

# Connaissances pour enseigner l'informatique : analyse textuelle de productions d'enseignants de l'école primaire

## 9

Gabriel PARRIAUX<sup>1,2</sup>  
Christophe REFFAY<sup>3</sup>  
Béatrice DROT-DELANGE<sup>4</sup>  
Mehdi KHANEBOUBI<sup>1</sup>

1. Université Paris Cité, Laboratoire  
EDA, 75006 Paris, France

2. Haute École Pédagogique (HEP)  
du canton de Vaud, 1007 Lausanne,  
Suisse

3. Université de Franche-Comté,  
ELLIADD, 25000 Besançon, France

4. Université Clermont-Auvergne,  
INSPÉ Clermont-Auvergne, Labo-  
ratoire ACTÉ, 63400 Chamalières,  
France

Dans ce chapitre, nous portons nos analyses sur ce que des documents pédagogiques (scénarios pédagogiques, fiches de préparation d'activités, retours d'expérience sur une activité mise en œuvre) pour l'enseignement de l'informatique, produits par des enseignants novices, peuvent révéler de leurs connaissances didactiques. Partant d'un ensemble de documents produits par les étudiants – futurs enseignants de trois instituts de formation en France et en Suisse, nous rendons compte d'un parcours de recherche en quatre étapes.

Dans une première étude (Drot-Delange *et al.*, 2021), ces productions ont été inventoriées et classées manuellement selon l'institution d'origine, le niveau des élèves ciblés (entre 6 et 11 ans), le type d'activités proposées (informatique débranchée, programmation, jeux sérieux ou robotique pédagogique), le matériel utilisé (robot ou application spécifique), les connaissances ou compétences ciblées (en informatique ou non) et les éventuels concepts informatiques manipulés. Dès lors, nous avons relevé dans quelques-unes de ces productions ce qui, d'un point de vue didactique, nous semble s'apparenter à des mécompréhensions, voire des manques. Deux de ces erreurs sont apparues plusieurs fois dans notre corpus et sont en lien avec un usage de vocabulaire informatique peu approprié, voire erroné, selon le contexte. La première consiste en une confusion au niveau des termes utilisés pour parler d'une donnée, en tant que représentation statique d'une information, ou d'une instruction, en tant qu'élément dynamique dans un programme, qui peut être exécuté. La seconde consiste en une utilisation erronée de vocabulaire tiré du paradigme de programmation séquentielle dans un contexte de programmation événementielle, en particulier dans des activités de robotique avec Thymio. Ce constat nous amène à penser que les spécificités des robots pédagogiques et leurs liens avec différents paradigmes



de programmation devraient être davantage explicités dans les formations d'enseignants.

Lors d'une seconde étude (Reffay *et al.*, 2023), afin de mieux investiguer cette dernière problématique, nous avons restreint le corpus aux seuls documents portant sur des activités de robotique, puis y avons ajouté des productions d'enseignants plus expérimentés, recueillies sur le web, en vue d'effectuer des comparaisons. Ayant inventorié le lexique de ce nouveau corpus à l'aide de méthodes d'analyse textuelle, nous en avons extrait les mots en lien avec l'informatique et avons pu vérifier la manière dont le lexique, ainsi que les notions informatiques qu'il véhiculait, différaient pour chaque robot.

Une troisième étude (Parriaux *et al.*, 2023) est venue prolonger la seconde, toujours sur le même corpus de robotique éducative. Adoptant une approche plus globale, considérant la totalité du lexique présent dans le corpus, nous avons extrait les thématiques à l'aide d'une méthode de *clustering*. Avec cette technique inductive et partiellement automatisée, il a ensuite été possible de montrer l'association qui existait en termes de lexique entre chacun des robots et l'une de ces thématiques.

Ce chapitre intervient comme la quatrième et dernière étape de ce parcours de recherche. Nous revenons au corpus de départ constitué de productions incluant une diversité d'activités débranchées ou non, des outils, comme les robots, mais aussi différents logiciels d'apprentissage de la programmation. Nous y appliquons les méthodes d'analyse textuelle de l'étude précédente, comme le *clustering*, afin de déterminer de manière inductive les thématiques abordées dans ces documents, ainsi que les liens entre le lexique et le type d'outil mobilisé ou le degré scolaire visé.

## Cadre théorique et questions de recherche

Nous nous intéressons aux connaissances didactiques en lien avec l'enseignement de l'informatique. Par connaissances didactiques, nous entendons les connaissances professionnelles des enseignants qui sont spécifiques à l'enseignement d'un contenu donné, dans notre cas l'informatique. Nous considérons qu'un moyen d'accéder à ces connaissances, c'est d'en chercher les traces par exemple dans les productions des enseignants.

Pour approfondir ce que nous entendons par « connaissances didactiques », nous inscrivons notre étude au sein du cadre conceptuel des connaissances pédagogiques du contenu ou *Pedagogical Content Knowledge* (PCK) proposé par Shulman (1986, 2007) et enrichi par Magnusson *et al.* (1999). Ces derniers le décrivent comme constitué de cinq composants : la connaissance

des curricula, la connaissance de la compréhension des élèves, la connaissance des stratégies d'enseignement, la connaissance de l'évaluation ainsi que l'orientation en matière d'enseignement de la discipline. Ce dernier composant se réfère aux connaissances et aux croyances d'un enseignant envers les finalités de l'enseignement d'une discipline à un certain degré scolaire.

Plus récemment, une communauté de chercheurs en didactique des sciences a poursuivi le développement de ce cadre théorique pour le formaliser dans un « modèle de consensus affiné » ou *Refined Consensus Model* (RCM) des PCK (Carlson *et al.*, 2019). Ce nouveau modèle a la particularité de distinguer trois domaines de PCK : le PCK collectif (cPCK), le PCK personnel (pPCK) et le PCK en acte *enacted PCK* (ePCK). Ce dernier domaine comprend l'ensemble des connaissances qui se manifestent dans un contexte précis de l'activité de l'enseignant auprès d'une classe déterminée d'élèves, sur un sujet particulier et dans un contexte bien défini. Le ePCK déborde du lieu et du temps de la classe pour concerner également les activités en lien avec la planification et l'évaluation d'une leçon.

Plusieurs traductions en français cohabitent pour l'expression *Pedagogical Content Knowledge* (Kermen et Izquierdo-Aymerich, 2017). Nous choisissons la traduction en « connaissances didactiques » qui nous semble la plus accessible en français et la plus proche de la signification originale de Shulman.

Nos questions de recherche sont les suivantes :

1. QR1 : que pouvons-nous dire des thématiques abordées par les enseignants dans les ressources pédagogiques qu'ils ont produites pour enseigner l'informatique ?
2. QR2 : qu'est-ce que l'analyse du lexique composant ces ressources nous apprend au sujet des connaissances des enseignants pour enseigner l'informatique ?
3. QR3 : quelles sont les relations que l'on peut établir entre le lexique présent dans les ressources, le type d'activités, les outils et l'âge des élèves concernés ?

## Corpus et méthode

### Présentation du corpus

Notre corpus est constitué d'un total de 140 ressources pédagogiques pour enseigner l'informatique produites par des étudiants – futurs enseignants du primaire en formation initiale dans les Instituts nationaux supérieurs du professorat et de l'éducation (INSPE) de Clermont Auvergne et de Besançon en France, ainsi qu'à la Haute école pédagogique (HEP) du canton de Vaud en

Suisse, entre 2019 et 2020. Ces ressources, ainsi que les connaissances dont elles sont le recueil, sont influencées par de multiples facteurs, parmi lesquels le contexte de formation dans lequel elles ont été produites. Issues de formations initiales de futurs enseignants, elles rendent compte de la création par les étudiants de scénarios pédagogiques pour enseigner la science informatique à l'école. Pour l'INSPE de Besançon, il s'est agi de devoirs donnés aux étudiants dans le cadre d'un module obligatoire du master métiers de l'enseignement de l'éducation et de la formation (MEEF). Pour l'INSPE de Clermont Auvergne, ce sont des comptes rendus d'activités de programmation visuelle réalisées dans le cadre du module certificat informatique et Internet niveau 2 (C2i2e) du master MEEF. Pour la HEP du canton de Vaud, il s'est agi de présentations effectuées pour un examen oral validant un module facultatif d'introduction à la science informatique. Ces scénarios étaient en général réalisés par deux et étaient parfois expérimentés en classe.

De ce fait, il est tout à fait imaginable que le contenu de ces ressources ait subi l'influence du contexte de formation et des exigences institutionnelles formulées aux étudiants. Dans le cas de l'INSPE de Besançon, les étudiants étaient amenés à produire leur séquence à l'aide d'un outil auteur qui, par son format et ses rubriques, impose l'adjonction de certaines informations. Dans le cas de la HEP Vaud, les étudiants recevaient des critères précis d'évaluation de leur présentation qui les incitait à traiter de certains contenus. Nous pouvons notamment penser que la définition d'objectifs d'apprentissage, en lien avec les programmes scolaires, faisait partie des contraintes formulées. Nous pouvons également nous questionner sur le fait de savoir si nous aurions trouvé les mêmes éléments dans des ressources d'enseignants produites hors d'un tel contexte de formation. L'impact du contexte sur les résultats n'est donc pas totalement négligeable.

Nous précisons ci-après les variables catégorielles associées à ces ressources avec leurs différentes modalités. Pour chaque modalité de variable, nous précisons entre parenthèses le nombre de ressources associées à cette modalité. Certaines ressources peuvent contenir plusieurs activités et peuvent donc être associées à plusieurs modalités d'une même variable.

- Domaine informatique : algorithmique-programmation (131) et représentation des données (14).
- Type d'activité : informatique débranchée (117), robotique (76), programmation visuelle (39) et micromonde (19) (application incluant consignes, aides, évaluation et retours automatiques permettant à l'élève de s'exercer en autonomie).
- Outil : jeu du robot (74), déplacement sur grille (61), Bluebot (29), BeeBot (22), Thymio (17), ScratchJr (27), Scratch (15),

cryptographie (10). Dix-sept autres outils qui apparaissaient de manière très ponctuelle ( $n \leq 5$ ) n'ont pas été retenus dans les analyses.

- Genre des groupes d'étudiants : masculin (6), féminin (108) et mixte (26).
- Formation suivie : huit heures (82) (INSPE de Besançon), 28 heures (46) (HEP Vaud) et 4 heures (12) (INSPE de Clermont Auvergne).
- Niveau scolaire des élèves : élémentaire (41) (3-6 ans), primaire inférieur (58) (6-9 ans) et primaire supérieur (50) (9-11 ans).

En dehors des niveaux scolaires très également répartis, les effectifs (à plat) de ces modalités montrent des disparités importantes pour chaque variable : davantage de programmation (90 %) que de représentation des données (10 %), une forte proportion d'activités débranchées (47 %) avec le jeu du robot<sup>1</sup> (25 %) et les déplacements sur grille (21 %), ainsi que de la robotique (30 %). Parmi les sept robots mentionnés dans notre corpus, les plus utilisés sont : Bluebot (37 %), BeeBot (28 %) et Thymio (22 %).

Les documents produits sont issus de formations ayant des durées très variables : 59 % sont issus de la formation de Besançon (8 h), 33 % de Lausanne (28 h) et 9 % de Clermont (4 h). Ce déséquilibre est encore accentué par le fait que les productions de Lausanne ont pris la forme de présentations orales dans le cadre d'un examen. Leur contenu est donc, par le format de diapositives et la transcription de l'audio, plus court en termes de texte que le contenu des documents plus détaillés constitués à Besançon et à Clermont où ils prenaient la forme de scénarios pédagogiques déroulés sur plusieurs pages et incluant les objectifs pédagogiques, les ressources utilisées, le rôle de l'enseignant et des élèves et les différentes activités à mettre en œuvre.

## Analyse lexicale et textométrie

Nous choisissons d'analyser notre corpus à l'aide de méthodes issues du domaine de l'analyse des données textuelles (ADT). Leur particularité consiste à appliquer une approche statistique à l'étude du texte (Lebart *et al.*, 2019).

Cette méthode statistique considère le texte comme un ensemble de mots que l'on dénombre pour en analyser le contenu. La fréquence d'apparition de mots du lexique peut être comparée à une fréquence attendue dans le cas d'une distribution aléatoire de vocabulaire, et cet écart à la moyenne, mis en relation avec les caractéristiques des ressources observées, permet de mettre en évidence des proximités ou au contraire des oppositions entre certaines de ces caractéristiques et les termes du lexique.

1. Le jeu du robot est une activité d'apprentissage de la programmation dans laquelle un élève, ou l'enseignant, joue le rôle d'un robot qui doit être guidé par des instructions données par le reste de la classe.

## Classification de Reinert

L'algorithme de classification de Reinert permet de regrouper des segments d'un corpus dans des catégories qui se distinguent par le lexique utilisé (Reinert, 1983). Par le biais de cette classification ou *clustering*, le chercheur voit surgir des thématiques qui constituent l'univers linguistique investi par les individus ayant généré les données du corpus. Cette méthode doit donc nous permettre de répondre à la première question de recherche (QR1).

Cet algorithme est fondé sur une classification descendante hiérarchique<sup>2</sup>. De manière itérative, il divise le corpus en deux clusters puis va identifier un maximum d'hétérogénéité en optimisant l'homogénéité de chaque cluster. Cette opération est répétée jusqu'à ce que le nombre de clusters demandé par le chercheur soit atteint. Il s'agit d'une méthode de classification non supervisée, ce qui signifie que les catégories ne sont pas connues à l'avance et sont déterminées par le corpus lui-même. C'est la raison pour laquelle nous parlons ici d'une méthode inductive.

Dans notre cas, le prétraitement des données et les analyses sont conduits dans le langage R (R Core Team, 2022)<sup>3</sup>. Nous procédons à une segmentation du corpus en segments d'environ 40 mots, puis nous réalisons sa lemmatisation, à savoir la réduction de ses termes à leur forme générique, à l'aide de la bibliothèque *SpacyR* dans R et le modèle de langage *fr\_dep\_news\_trf* pour le français. Le corpus est ainsi découpé en *tokens* avec séparation aux espaces. Les mots vides (ne véhiculant pas de sens) sont retirés ainsi que les termes apparaissant dans moins de trois segments (considérés non significatifs).

Une fois ces prétraitements effectués, le lexique est extrait des segments et une matrice est constituée, appelée matrice termes-documents, avec, pour colonnes, les termes du lexique et en lignes, les segments. Chaque cellule, à l'intersection d'une ligne (segment) et d'une colonne (terme), indique alors la présence d'un terme du lexique par un 1 lorsqu'il apparaît (une ou plusieurs fois) dans le segment ou son absence par un 0.

La méthode de Reinert cherche à maximiser la distance du  $\chi^2$  entre deux regroupements de segments. Pour choisir les segments à regrouper, Max Reinert propose d'effectuer une analyse factorielle des correspondances (AFC) de la matrice termes-documents, puis d'ordonner les segments en fonction de leurs coordonnées sur le premier axe issu de cette AFC, c'est-à-dire celui qui maximise les oppositions entre les segments. Pour réaliser la scission du corpus en deux clusters, Reinert réalise alors différents regroupements en se fondant sur cet ordre-là en cherchant toujours à maximiser la distance du  $\chi^2$  entre les deux clusters. Cette étape est reproduite, de manière itérative, jusqu'à obtenir le nombre de clusters souhaité.

2. Les méthodes de classification descendante hiérarchique (CDH) constituent une famille de méthodes de classification qui permettent de trier une population en différentes classes sur la base de critères de ressemblance entre individus. La CDH procède par bipartitions successives, c'est-à-dire que l'on part du haut vers le bas : on commence par considérer toute la population dans son ensemble, puis on la divise en deux classes, et ce de manière itérative jusqu'au nombre de classes souhaité. Ces méthodes se différencient des techniques de classification ascendante hiérarchique (CAH) qui, à l'inverse, fonctionnent du bas vers le haut : partant des individus considérés individuellement les uns des autres, on constitue une classe en regroupant les deux individus les plus ressemblants. De manière itérative à nouveau, ces regroupements sont opérés jusqu'à obtenir un nombre de classes qui est pertinent pour l'interprétation du chercheur.

3. R est un langage de programmation particulièrement utilisé dans le domaine des analyses de données quantitatives.

Chaque cluster est caractérisé par le nombre de segments qui le composent et la liste des termes qui s'y trouvent surreprésentés. Ces termes, ainsi que le contexte des segments dans lesquels ils apparaissent, sont ceux qui permettent d'interpréter le sens que prend le cluster et d'en déduire la thématique à laquelle il se rattache.

## Analyse des correspondances

Une fois le lexique extrait, nous nous intéressons aux relations entre celui-ci et les différentes caractéristiques de nos ressources.

En observant la présence de termes spécifiques dans certains documents et pas dans d'autres, ou leur fréquence d'apparition plus élevée dans certaines parties de corpus que dans d'autres, et en croisant cette information avec les caractéristiques des différents documents ou parties de documents, il est possible d'établir des proximités entre certains mots du lexique et certaines ressources. Pour mener une telle analyse, il est nécessaire de disposer d'une méthode qui permet de visualiser ces distances entre lexique et ressources, ou caractéristiques des ressources, et c'est ici qu'intervient l'analyse des correspondances.

L'analyse des correspondances (Benzécri, 1973) est une méthode statistique exploratoire utilisée pour analyser les relations entre des variables catégorielles. Son objectif est de mettre en évidence les relations entre les différentes modalités des variables étudiées et de les représenter graphiquement. On peut considérer que l'analyse factorielle des correspondances multiples est homologue de l'analyse en composantes principales appliquée à des variables catégorielles.

À partir du tableau de contingence, dont la composition est précisée plus loin, l'algorithme va procéder à une réduction de la dimensionnalité des données et permettre leur projection sur un espace à deux dimensions. Le graphique obtenu (tel que celui de la figure 2) permet de visualiser les oppositions ou au contraire les rapprochements entre les différentes catégories étudiées (Beaudouin, 2016).

Dans le contexte de l'ADT, l'analyse des correspondances permet une représentation conjointe des termes et des variables dans un plan factoriel qui montre les attractions entre les termes du lexique et les modalités de variables.

## Résultats et discussion

### QR1 : quels sont les thèmes abordés dans notre corpus ?

Les prétraitements appliqués à notre corpus nous permettent d'obtenir une matrice termes-documents qui croise 2407 termes et 5358 segments. Le *clustering* de Reinert est réalisé dans R à l'aide de la bibliothèque Rainette. Après plusieurs essais, pour éviter des clusters trop volumineux ( $n > 1000$ ), nous décidons de diviser notre corpus en un total de 34 clusters.

Ainsi que le présente la figure 1, nous constatons que la taille des clusters produits est très inégale.

Pour la suite des analyses, nous décidons de ne retenir que les dix premiers clusters contenant plus de 200 segments et présentés ci-après par ordre décroissant du nombre de segments. Pour chacun d'entre eux, nous avons analysé les segments où apparaissent les termes surreprésentés pour proposer un titre. Les cinq premiers mots surreprésentés de chaque cluster sont indiqués entre parenthèses après la description du cluster.

- Cluster 9 — déplacements avec directions : description d'activités de déplacement, parfois avec appui sur des touches, pour effectuer diverses actions comme tourner, avancer. Des directions sont indiquées, le référentiel est relatif au robot ou personnage qui se déplace (avancer, tourner, action, droite, touche).
- Cluster 11 — déplacements sur des cases : description d'activités de déplacement sur quadrillage, depuis une case de départ à un point d'arrivée en suivant un certain parcours. Le référentiel est absolu et ne fait pas référence à la position du personnage ou du robot (case, arrivée, départ, point, fleur).
- Cluster 4 — séquence pédagogique, contexte et apprentissage : présentation d'une séquence pédagogique composée de plusieurs leçons avec une mise en contexte (culture numérique, monde numérique, science informatique) et des apprentissages qui y sont liés (séquence, numérique, apprentissage, printemps, discipline).
- Cluster 17 — institutionnalisation : moments collectifs de mise en commun, d'institutionnalisation (commun, mise, institutionnalisation, collectif, réponse).
- Cluster 19 — activité enseignants et élèves : description de l'activité des enseignants et des élèves et de leurs interactions ; explicitation des consignes, organisation du travail seul, en binôme (élève, enseignant, recherche, consigne, groupe).

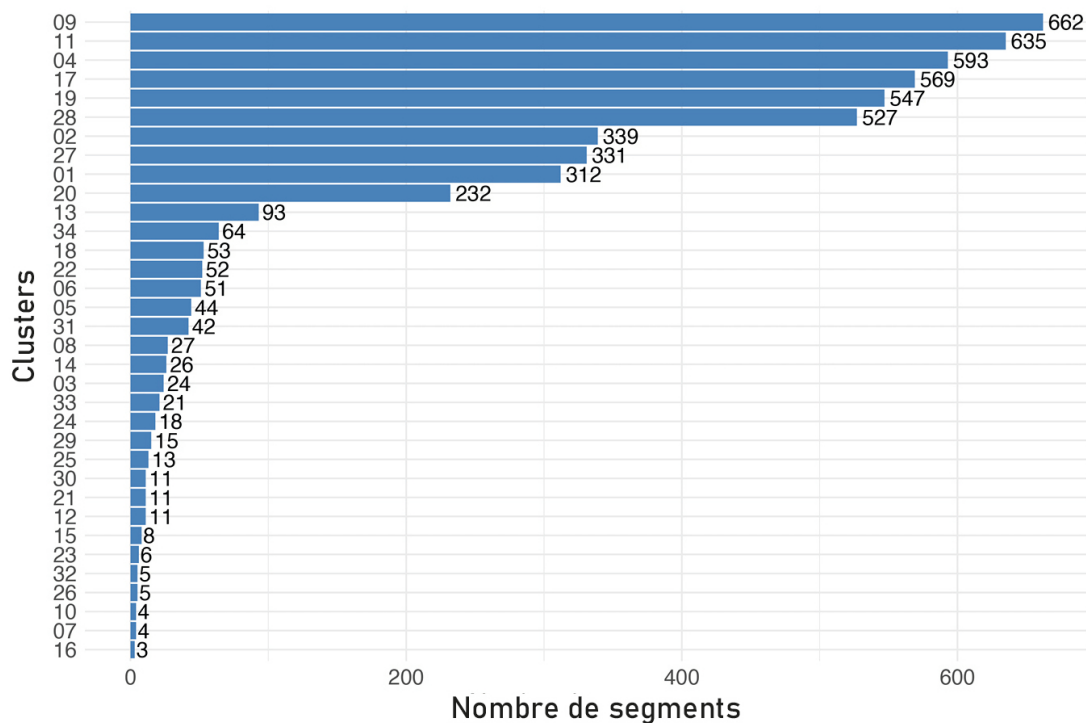


FIGURE 1 – Les 34 clusters de la classification de Reinert et leurs tailles.

- Cluster 28 — séance, phase, matériel et objectifs : segments en lien avec la présentation d'une ou de plusieurs séances d'enseignement avec le matériel, le découpage en phases d'un certain nombre de minutes, les objectifs et l'évaluation (séance, matériel, minute, phase, objectif).
- Cluster 2 — domaines et compétences du socle : éléments du socle de compétences, notamment dans le domaine scientifique et mathématique, en référence à une leçon. Liens avec le langage, la communication (domaine, compétence, exprimer, socle, scientifique).
- Cluster 27 — rappels et réinvestissement : activités de réinvestissement, rappel de la séance précédente; mention de la tablette soit dans une liste de matériels, soit dans la description d'une activité; logiciel ScratchJr ou autres logiciels et applications (précédent, rappel, tablette, réinvestissement, séance).
- Cluster 1 — cycle et repérage dans l'espace : segments qui font mention du ou des cycle(s) d'enseignement pour lesquels ces activités sont prévues. Ils font également allusion au repérage dans l'espace ou le plan ainsi qu'à des déplacements qu'il s'agit de décrire (cycle, espace, repérer, relation, familier).



- Cluster 20 — difficultés, autonomie et remédiation : gestion des difficultés des élèves par l'enseignant, remédiation; différenciation et autonomie (difficulté, auprès, autonomie, travail, moment).

L'interprétation du sens de ces clusters nous permet de répondre à notre première question de recherche : que pouvons-nous dire des thématiques abordées par les enseignants dans les ressources pédagogiques qu'ils ont produites pour enseigner l'informatique ?

- La première thématique se rapporte à l'échelle de la séquence, en tant qu'enchaînement de séances. Les segments qui s'y rattachent se situent plutôt tout au début des ressources desquelles ils proviennent. Ils donnent le contexte général et mentionnent les apprentissages visés (cluster 4). Nous situons cette thématique dans une granularité plus grossière que les suivantes et la nommons « contexte général d'une séquence ».
- Une deuxième thématique concerne l'échelle de la leçon, donc avec une granularité plus fine que la précédente, et est en lien avec les domaines et compétences de la séance, ses phases, son matériel et ses objectifs, ainsi qu'avec les activités de l'enseignant et des élèves (clusters 2, 19 et 28). Nous l'intitulons « caractéristiques d'une leçon ».
- Une troisième thématique est constituée par les moments clés d'une leçon autour de l'institutionnalisation, de la gestion des difficultés et de la remédiation, ainsi que les rappels et réinvestissements (clusters 17, 20 et 27). Nous lui donnons le nom de « moments clés d'une leçon ».
- Une quatrième thématique est celle des déplacements, typique des activités d'apprentissage de la programmation chez des enfants, principalement sous forme d'activités débranchées (jeu du robot) ou de robotique (BeeBot, Bluebot) (clusters 9 et 11). Nous la nommons « notions de déplacement ».

À noter que le cluster 1 – cycle et repérage dans l'espace – est un peu à la frontière des première, seconde et quatrième thématiques, puisqu'il mentionne des éléments plutôt généraux en lien avec la séquence, d'autres plus spécifiquement en lien avec la séance, et d'autres encore en lien avec le repérage et les déplacements.

Nous avons donc quatre thématiques que l'on pourrait assimiler à différents plans de cadrage sur l'objet d'enseignement : un plan d'ensemble à l'échelle de la séquence composée de plusieurs leçons, un plan moyen à l'échelle d'une leçon ; deux plans rapprochés sur les moments clés d'une leçon et sur les notions de déplacements.

## QR2 : qu'est-ce que l'analyse du lexique composant ces ressources nous apprend des connaissances des enseignants ?

Les résultats de nos analyses des ressources pédagogiques montrent que les enseignants utilisent davantage de termes informatiques dans les parties en lien avec des savoirs institutionnalisés que dans celles en lien avec des connaissances contextualisées.

Nous allons expliciter ce résultat en explorant plus avant la signification de chacune des thématiques, ainsi que des clusters qui s'y rattachent, par l'analyse des segments qui les composent et par les termes surreprésentés.

Les termes assimilés à des notions informatiques sont surreprésentés dans la première thématique portant sur l'échelle de la séquence cluster 4 — séquence pédagogique, contexte et apprentissage<sup>4</sup> et en partie cluster 1 — cycle et repérage dans l'espace<sup>5</sup>. En dehors de cette thématique, très peu de termes informatiques apparaissent ailleurs. On trouve le terme de « boucle » dans la quatrième thématique des déplacements (cluster 9 — déplacements avec directions).

Les noms des outils employés figurent parmi les termes surreprésentés des thématiques des déplacements et des moments clés (clusters 11, 27 et 20) : « BeeBot », « ScratchJr » et « Bluebot » respectivement. Il peut sembler étonnant que le terme « BeeBot » soit associé au cluster 11, en lien avec des déplacements dans un référentiel absolu, alors qu'on aurait plutôt tendance à l'associer à un référentiel relatif que l'on retrouve dans le cluster 9, au vu de la manière dont ce robot se programme avec des boutons gauche-droite, notamment. 15 % des segments du cluster 11 contiennent le terme « BeeBot », contre 4 % seulement du cluster 9. De façon similaire, l'association entre le terme « BeeBot » et le cluster 20 — difficultés, autonomie et remédiation surprend. Nous faisons l'hypothèse qu'elle est due plutôt au contexte de formation qu'à une spécificité de ce robot.

Ces analyses apportent des éléments de réponse à notre seconde question de recherche : qu'est-ce que l'analyse du lexique composant ces ressources nous apprend au sujet des connaissances des enseignants pour enseigner l'informatique ?

Les ressources pédagogiques produites par des enseignants constituent des traces de leurs connaissances didactiques-en-acte (ePCK). L'analyse de leur lexique nous montre que les enseignants utilisent davantage de termes informatiques lorsqu'ils sont dans une vision éloignée de l'activité de terrain, de l'acte d'enseigner. C'est dans la première thématique « contexte général d'une séquence » et du cluster 4 — séquence pédagogique, contexte et apprentissage, qui traitent de la séquence d'enseignement sous forme d'une

4. Avec les termes informatique, programmation, cryptage, machine, technologie, décryptage, algorithme, débrancher.

5. Avec les termes algorithmique, programme, coder, programmer, automatique.

introduction et d'une mise en contexte d'une série de leçons, que le plus grand nombre de termes informatiques apparaissent. Souvent, les termes font partie d'expressions reprises à l'identique d'objectifs inscrits dans les textes officiels. Selon le cadre conceptuel des PCK, les éléments tirés des programmes ont à voir avec la connaissance du curriculum, qui recouvre des connaissances institutionnalisées. Celles-ci peuvent également être vues comme des connaissances didactiques collectives ou *collective PCK* (cPCK), une base de connaissance spécialisée qui est partagée par un groupe de professionnels (Carlson *et al.*, 2019).

Notre étude montre que lorsque l'on se rapproche du niveau de la séance elle-même, du détail des activités de classe et de leurs spécificités, comme dans les deuxième et troisième thématiques, « caractéristiques d'une leçon » et « moments clés d'une leçon », les termes informatiques ressortent beaucoup moins. Cela ne veut pas dire qu'ils n'existent pas ou n'y sont pas du tout présents, mais qu'ils ne sont pas surreprésentés dans l'un ou l'autre des clusters constituant ces parties de productions. Selon le cadre des PCK, nous rapprocherions ces thématiques du niveau des ePCK, connaissances didactiques-en-acte, définies comme les connaissances spécifiques d'un enseignant dans un contexte particulier, avec un groupe d'élèves déterminé ayant pour but d'apprendre un concept ou un aspect particulier d'une discipline (Carlson *et al.*, 2019).

Nous voyons une possible convergence dans la distinction entre les connaissances didactiques collectives et en-acte (cPCK *vs.* ePCK) du modèle de consensus affiné des PCK (Carlson *et al.*, 2019) et la distinction savoir/connaissance du cadre de la didactique française des mathématiques (Margolinas, 2014). Pour celle-ci, les savoirs existent en tant que texte écrit et constituent une construction sociale, partagée, validée. On les retrouve, par exemple, dans les curricula des différentes disciplines. Les connaissances, elles, sont davantage personnelles et incarnées. Elles représentent une mise en contexte d'un savoir institutionnel par un acteur dans une situation bien définie.

Dans notre corpus, les portions de textes en lien avec la première thématique (cluster 4 — séquence pédagogique, contexte et apprentissage) portant un regard d'ensemble sur la séquence contiennent un certain nombre de notions informatiques et peuvent être rapprochées de savoirs ou de connaissances didactiques collectives (cPCK), tandis que les portions de texte portant un regard rapproché sur la leçon ne contiennent que peu de notions informatiques et sont à rapprocher de connaissances ou de connaissances didactiques-en-acte (ePCK).

Pour ces raisons, nous pouvons dire que les enseignants utilisent davantage de notions informatiques lorsqu'ils considèrent la séquence dans son ensemble et que l'on a plutôt à faire à des savoirs

institutionnalisés. Ils en utilisent moins lorsqu'ils se rapprochent du détail de la leçon et que ce sont des connaissances contextualisées qui sont en jeu.

### **QR3 : quelles relations entre le lexique et les modalités de variables du corpus ?**

Pour répondre à notre troisième question de recherche (QR3), nous opérons une analyse des correspondances sur les données lexicales tirées de notre corpus. Nous souhaitons observer les relations qu'il peut y avoir entre les mots du lexique et les caractéristiques des productions. Compte tenu du nombre plus important de termes dans les productions issues de la formation qui a duré 8 h (Besançon) et des biais que cela risquerait de générer au niveau des interprétations, nous décidons de ne pas tenir compte de la variable liée au type de formation suivie.

Nous constituons donc un tableau lexical des questions, selon la méthode décrite par Cibois (1990), qui croise les mots du lexique avec l'ensemble des modalités de nos variables. Il est constitué des 21 modalités de variables en lignes et des 2407 mots du lexique en colonnes. Chaque cellule compte le nombre d'occurrences du mot du lexique dans les productions associées avec la modalité de la variable concernée. Nous avons conservé les variables : « domaine informatique », « genre », « type d'activité », « outil » et « niveau scolaire des élèves ».

Nous appliquons l'analyse des correspondances sur un tableau de 18 modalités et 77 termes. Les deux premiers axes produits par l'analyse des correspondances représentent une inertie cumulée de 56 %.

Nous présentons sur la figure 2 les associations et oppositions apparentes du premier plan factoriel.

Lors d'une analyse des correspondances, l'interprétation se base sur les distances entre les objets. Les proximités au niveau des positions sur le plan rendent compte de potentielles associations entre des termes ou des caractéristiques des ressources. Au contraire, des positions opposées par rapport aux axes représentent des oppositions au niveau du lexique ou des variables.

Dans une première lecture correspondant à l'axe 1 (horizontal) de l'analyse des correspondances, les activités de robotique et le jeu du robot sont plutôt en lien avec le niveau scolaire élémentaire et un vocabulaire de découverte, alors que les activités de type micromonde et de programmation visuelle visent plutôt les niveaux scolaires supérieurs et sont en lien avec un vocabulaire de déplacement, de repérage, d'informatique et de mathématique.

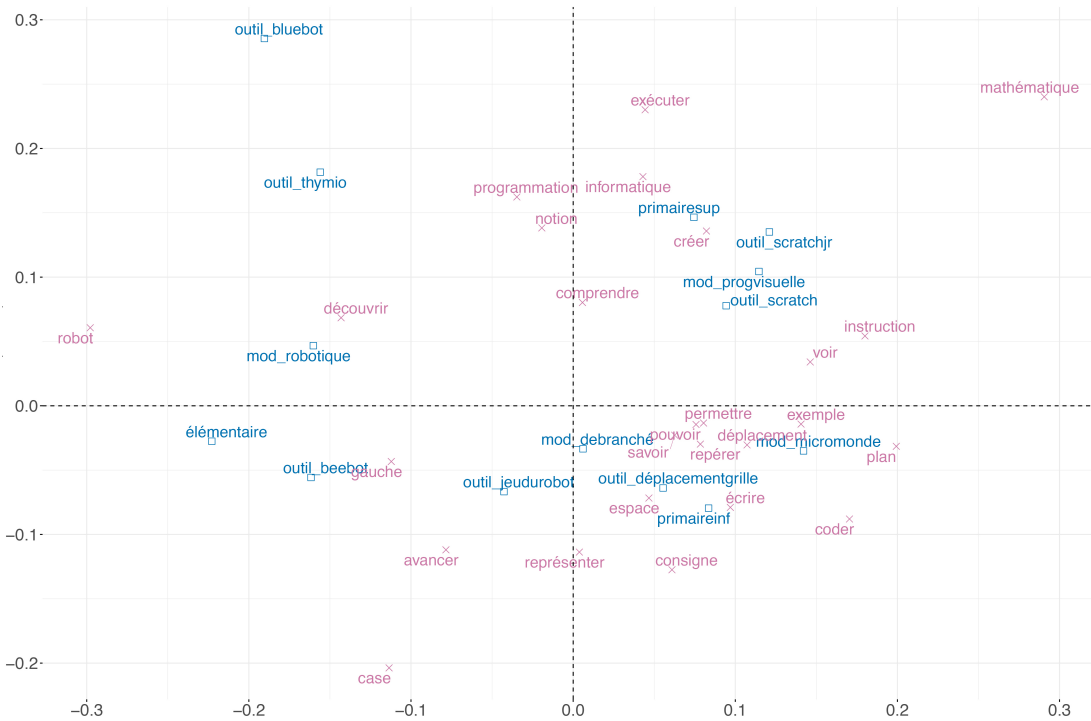


FIGURE 2 – Analyse des correspondances croisant le lexique et les variables. Le lexique est en violet avec des puces en croix ; les variables sont en bleu avec des puces carrées. Les 14 modalités de variable et les 26 termes du lexique avec les plus hautes valeurs de  $\cos^2$  sont représentés.

Selon une seconde lecture correspondant à l'axe 2 (vertical) de l'analyse des correspondances, les activités de robotique et de programmation visuelle sont plutôt mises en lien avec le niveau scolaire du primaire supérieur et un vocabulaire informatique, alors que les activités débranchées sont plutôt en lien avec les niveaux scolaires inférieurs et un vocabulaire de déplacement.

L'axe 1 oppose la robotique dans les classes de niveau élémentaire et les autres activités de type programmation visuelle et micromonde avec des élèves plus âgés.

À gauche de l'axe, on trouve le type d'activité robotique, le niveau scolaire élémentaire (3-6 ans), les outils Thymio et BeeBot, le jeu du robot. À droite de l'axe se situent les types d'activité micromonde et programmation visuelle, le niveau de scolarité primaire inférieur (6-9 ans), les déplacements sur grille, les outils ScratchJr et Scratch.

Au niveau du lexique, à la robotique sont associés les termes « robot », « découvrir » et « gauche ». Aux activités de programmation visuelle et micromonde sont associés les termes « coder », « déplacement », « instruction », « plan », « savoir », « repérer » et « mathématique ».

L'axe 2 oppose quant à lui la programmation visuelle et la robotique avec des élèves de la fin du primaire aux activités débranchées avec des élèves plus jeunes.

Au-dessus de l'axe, on trouve les outils Bluebot, Thymio et ScratchJr, le niveau scolaire primaire supérieur (9-11 ans) et le type d'activité programmation visuelle. Au-dessous de l'axe, on trouve le type d'activité débranché, les déplacements sur grille et le jeu du robot, ainsi que le niveau scolaire primaire inférieur.

En ce qui concerne le lexique, les termes « notion », « programmation », « comprendre », « informatique », « exécuter », « créer », « programme », « machine » sont à proximité de la programmation visuelle, alors que les termes « consigne », « représenter », « case », « espace », « parcours » et « avancer » sont du côté des activités débranchées.

Il est à noter que les variables de genre et de domaine de l'informatique ne ressortent pas de ces analyses et ne constituent pas des facteurs de différenciation en lien avec le lexique utilisé dans nos ressources.

## Conclusion

Cette recherche constitue la quatrième et dernière étape d'un processus d'analyse de ressources pédagogiques pour enseigner l'informatique produites par des étudiants-futurs enseignants de l'école primaire. Ces ressources constituent des traces de leurs connaissances didactiques-en-acte et l'analyse de leur lexique nous donne à voir le contenu et l'organisation de ces connaissances.

Nous avons appliqué des méthodes d'analyse des données textuelles telles que la classification de Reinert et l'analyse des correspondances pour investiguer le lexique présent dans ces ressources, extraire les grandes thématiques abordées et établir des liens entre le lexique et certaines caractéristiques de notre corpus.

En ce qui concerne notre première question qui s'intéresse aux thématiques abordées par les enseignants auteurs de ces ressources, nous faisons le constat que nous pouvons catégoriser ces thématiques en fonction de leur granularité : une première thématique concerne la séquence pédagogique en tant que suite de séances (plan d'ensemble), une seconde thématique est située à une échelle plus fine de la séance (plan moyen). Deux autres thématiques sont en lien avec des moments clés d'une leçon et des activités de déplacement (plans rapprochés).

Pour répondre à notre deuxième question sur ce que ce lexique nous apprend au sujet des connaissances didactiques des enseignants pour enseigner l'informatique, nous faisons le constat

que les termes informatiques sont davantage présents dans les clusters qui sont éloignés de la granularité fine de la séance. C'est lorsqu'ils présentent le contexte général d'une séquence pédagogique que les enseignants mentionnent des termes informatiques, notamment en tant que savoirs institutionnalisés. Mais plus l'on se rapproche du niveau de la séance en classe et des connaissances contextualisées, moins ces termes apparaissent. Nous pouvons faire l'hypothèse que la présence des termes informatiques en tant que savoirs institutionnalisés représente une influence du contexte de la formation et que les étudiants – futurs enseignants ont répondu à une nécessité de formation de coller au curriculum prescrit. Ceci pourrait constituer une future piste d'investigation, notamment par l'analyse des supports de formation. En ce qui concerne l'absence de termes informatiques au niveau de la leçon, nous pouvons faire l'hypothèse que les enseignants de l'école primaire opèrent une sorte de transposition didactique-en-acte, simplifiant leur expression pour se rapprocher de ce qu'ils considèrent comme accessible aux élèves.

Dans les résultats des recherches antérieures (Drot-Delange *et al.*, 2021), des difficultés de compréhension des concepts informatiques de la part des enseignants ont été relevées. En explicitant davantage ces notions au plus près des activités mises en œuvre et pas seulement dans les objectifs généraux de la séquence, les enseignants novices seraient plus à même de repérer de possibles incohérences.

Enfin, la troisième question nous amène à considérer les relations que l'on peut établir entre le lexique présent dans les ressources, le type d'activités, les outils et le niveau scolaire des élèves. Nous faisons le constat que les activités de robotique, lorsqu'elles s'adressent à des élèves de niveau scolaire inférieur, sont plutôt orientées vers de la découverte. Les activités débranchées, elles, sont plutôt en lien avec des notions de déplacement. Les activités de programmation et de robotique, lorsqu'elles concernent des élèves de niveau scolaire supérieur, sont quant à elles les plus portées vers des notions informatiques. Ces résultats constituent autant de pistes de recherche pour la didactique de l'informatique et pour la formation des enseignants.

### Recommandations et leçons tirées

Au niveau didactique, les conclusions de cette étude montrant une surreprésentation de termes informatiques dans les portions de ressources éloignées de la leçon mettent en lumière l'absence relative de notions informatiques dans le niveau de granularité fine des activités en classe. Cela nous amène à nous interroger sur l'explicitation des concepts informatiques durant les activités avec les élèves : comment les élèves s'approprient-ils ces concepts si le vocabulaire spécifique n'est que peu sollicité ? Quels liens peut-on établir entre apprentissages des élèves et langage, vocabulaire et notions ?

En ce qui concerne les connaissances professionnelles des enseignants pour enseigner l'informatique, nous estimons que l'explicitation des concepts constitue un enjeu de formation : comment rendre ces concepts actionnables par les enseignants dans leurs activités, afin qu'ils puissent y faire référence non seulement dans un contexte de connaissances institutionnelles, mais également au cœur des activités avec leurs élèves ?

En termes méthodologiques, les résultats de cette étude nous amènent à penser que les techniques issues de l'analyse de données textuelles sont porteuses d'un potentiel non négligeable et qu'il serait intéressant d'en explorer davantage les possibilités dans le domaine de la recherche en éducation et plus particulièrement en didactique de l'informatique. Il serait souhaitable de pouvoir les expérimenter sur d'autres supports textuels, comme dans l'analyse de transcriptions d'entretiens, de discours d'enseignants ou d'élèves en classe.

## Références

Beaudouin, V. (2016). Retour aux origines de la statistique textuelle : Benzécri et l'école française d'analyse des données, *Journées internationales d'Analyse statistique des Données Textuelle*, Nice, France, p. 17-27. <https://hal.science/hal-01376938>

Benzécri, J. P. (1973). *L'analyse des données, Tome 1 : La Taxinomie ; Tome 2 : L'analyse des correspondances*, Dunod.

Carlson, J., Daehler, K. R., Alonzo, A. C., Barendsen, E., Berry, A., Borowski, A., Carpendale, J., Kam Ho Chan, K., Cooper, R., Friedrichsen, P., Gess-Newsome, J., Henze-Rietveld, I., Hume, A., Kirschner, S., Liepertz, S., Loughran, J., Mavhunga, E., Neumann, K., Nilsson, P., (. . .) Wilson, C. D. (2019). « The refined consensus model of pedagogical content knowledge in science education », dans A. Hume, R. Cooper, et A. Borowski (dir.), *Repositioning Pedagogical Content Knowledge in Teachers' Knowledge for Teaching Science*, Springer, p. 77-94. [https://doi.org/10.1007/978-981-13-5898-2\\_2](https://doi.org/10.1007/978-981-13-5898-2_2)



Cibois, P. (1990). Éclairer le vocabulaire des questions ouvertes par les questions fermées : Le tableau lexical des questions, *Bulletin of Sociological Methodology / Bulletin de Méthodologie Sociologique*, vol. 26, n° 1, p. 12-21. <https://doi.org/10.1177/075910639002600102>

Drot-Delange, B., Parriaux, G., et Reffay, C. (2021). Futurs enseignants de l'école primaire : connaissances des stratégies d'enseignement, curriculaires et disciplinaires pour l'enseignement de la programmation, *Recherches en didactique des sciences et des technologies*, vol. 23, p. 55-76. <https://doi.org/10.4000/rdst.3685>

Kermen, I., et Izquierdo-Aymerich, M. (2017). Connaissances professionnelles didactiques des enseignants de sciences : un thème de recherche encore récent dans les recherches francophones, *Recherches en didactique des sciences et des technologies*, vol. 15, p. 9-32. <https://doi.org/10.4000/rdst.1479>

Lebart, L., Pincemin, B., et Poudat, C. (2019). *Analyse des données textuelles*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctvq4bxws>

Magnusson, S., Krajcik, J., et Borko, H. (1999). « Nature, sources, and development of pedagogical content knowledge for science teaching », dans *Examining pedagogical content knowledge*, Springer, p. 95-132. [https://doi.org/10.1007/0-306-47217-1\\_4](https://doi.org/10.1007/0-306-47217-1_4)

Margolinas, C. (2014). Connaissance et savoir. Concepts didactiques et perspectives sociologiques ? *Revue française de pédagogie. Recherches en éducation*, vol. 188, p. 13-22. <https://doi.org/10.4000/rfp.4530>

Parriaux, G., Reffay, C., Drot-Delange, B., et Khaneboubi, M. (2023). « Teachers' knowledge in informatics—Exploring educational robotics resources through the lens of textual data analysis », dans J.-P. Pellet et G. Parriaux (dir.), *Informatics in Schools. Beyond Bits and Bytes : Nurturing Informatics Intelligence in Education*, ISSEP 2023, *Lecture Notes in Computer Science*, vol. 14 296, p. 126-138. [https://doi.org/10.1007/978-3-031-44900-0\\_10](https://doi.org/10.1007/978-3-031-44900-0_10)

R Core Team, R. (2022). *R : A language and environment for statistical computing*.

Reffay, C., Parriaux, G., Drot-Delange, B., et Khaneboubi, M. (2023). « Robotics in primary education : A lexical analysis of teachers' resources across robots », dans T. Keane, C. Lewin, T. Brinda, et R. Bottino (dir.), *Towards a Collaborative Society through Creative Learning*, WCCE 2022, *IFIP Advances in Information and Communication Technology*, vol. 685. [https://doi.org/10.1007/978-3-031-43393-1\\_20](https://doi.org/10.1007/978-3-031-43393-1_20)

Reinert, M. (1983). Une méthode de classification descendante hiérarchique : application à l'analyse lexicale par contexte, *Cahiers de l'analyse des données*, vol. 8, n° 2, p. 187-198.

Shulman, L. S. (1986). Those who understand : Knowledge growth in teaching, *Educational researcher*, vol. 15, n° 2, p. 4-14. <https://doi.org/10.30827/profesorado.v23i3.11230>

Shulman, L.-S. (2007). Ceux qui comprennent. Le développement de la connaissance dans l'enseignement, *Éducation et didactique*, vol. 1, n° 1, p. 97-114. <https://doi.org/10.4000/educationdidactique.121>

Pour citer ce chapitre :

Parriaux, Gabriel, Reffay, Christophe, Drot-Delange, Béatrice, et Khaneboubi, Mehdi (2024). « Connaissances pour enseigner l'informatique : analyse textuelle de productions d'enseignants de l'école primaire », dans Cédric Fluckiger, Laetitia Boul'ch, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 173-191. <https://doi.org/10.53480/2024iecare09w>





# Former les enseignants à et par la programmation informatique cocréative en formation initiale

# 10

Frédérique LONGUET<sup>1</sup>  
Marcela Georgina GÓMEZ-ZERMEÑO<sup>2</sup>

1. Sorbonne Université, INSPE de l'Académie de Paris, 75017 Paris, France

2. Institut de recherche, d'innovation et d'études supérieures pour l'éducation, État de Nuevo León, Mexique

Ces dernières années, la majorité des gouvernements du monde entier se sont penchés sur l'intégration de la technologie comme un moyen de changement éducatif. Ceux qui promeuvent une plus grande utilisation du numérique dans l'éducation ont imaginé des écoles où les élèves acquièrent les « compétences du XXI<sup>e</sup> siècle » de manière personnalisée et collaborative, et à leur propre rythme avec des enseignants jouant le rôle de facilitateurs d'apprentissage (Voogt et Roblin, 2012). Or, l'éducation n'est pas seulement une question technique, mais aussi une clé pour vivre dans un monde en perpétuelle mutation. Pour comprendre les enjeux et soutenir une vision transformatrice de l'éducation, il nous semble nécessaire d'adopter une approche historique et contextualisée, en évitant le déterminisme technologique et le plaidoyer mal informé pour un passage d'une pédagogie anachronique et « analogique » à une pédagogie innovante et « numérique », fondée sur la reliance et le milieu (Longuet et Springer, 2021). C'est pourquoi nous sommes intéressées à une approche systémique de la programmation cocréative en formation initiale des enseignants. La programmation cocréative également appelée codéveloppement est une approche dans laquelle les programmeurs travaillent ensemble sur la création d'un programme informatique (*mob programming*). Cette approche diffère de la programmation individuelle traditionnelle, où un seul développeur écrit tout le code. Les enseignants stagiaires programmeurs collaborent étroitement, partagent des idées, résolvent des défis autour d'un seul écran. Cette programmation en équipe visant l'exploitation de la synergie entre les enseignants stagiaires conduit à des propositions didactiques et pédagogiques diverses en fonction des situations et des histoires personnelles.

Considérée comme une littérature du XXI<sup>e</sup> siècle, la programmation informatique est intégrée depuis 2016 dans les programmes scolaires en France. Bien que la programmation informatique occupe

une place significative dans le champ scolaire du cycle 1 à la terminale, elle ne s'inscrit pas dans une didactique de l'informatique dont les savoirs et savoir-faire seraient à maîtriser en formation initiale des enseignants. Elle est d'une part essentiellement arriérée aux mathématiques, aux sciences et à la technologie (Romero, 2016) et d'autre part définie dans le socle commun de connaissances, de compétences et de culture comme un langage parmi d'autres pour penser, partager des visions du monde (domaine 1 du socle) lors de situations cocreatives (domaine 2 et 3 du socle).

Des compétences transversales comme la créativité, la collaboration, la résolution de problèmes et la pensée critique sont de ce fait associables à la programmation pour que celle-ci ne soit pas réduite à des habiletés techniques (Romero *et al.*, 2017). La programmation cocreative semble donc être une clé pour répondre aux enjeux de l'éducation en termes de complexité, de diversité, d'interdisciplinarité et de créativité et une stratégie pour engager les enseignants et leurs élèves novices du 1<sup>er</sup> et 2<sup>e</sup> degrés issus de différentes disciplines dans une pensée complexe qui dépasse la programmation procédurale.

Dans ce cadre, notre question de recherche peut être formulée ainsi : quel est le rôle de la cocreativité dans le développement de la pensée complexe, dans l'acquisition de connaissances et compétences informatiques et transversales en formation initiale des enseignants ?

## Former à et par la programmation cocreative

### Opter pour une approche expérientielle

Considérer la programmation cocreative comme une ouverture à l'expérience en formation professionnelle, c'est s'inscrire à la fois dans une vision philosophique existentialiste qui considère la construction de l'identité personnelle et professionnelle comme une démarche active et anthropologique qui place l'être humain au cœur de la formation. Pour Dorais (2004), l'identité n'est pas pré-donnée, fixée ou extérieure mais construite à travers la confrontation personnelle du sujet à son environnement et aux autres. Elle est à la fois personnelle puisqu'elle résulte de l'intention du sujet à entrer en relation avec son environnement, et collective puisque les interactions avec l'environnement supposent le dialogue avec les autres sujets. La construction de l'identité personnelle et professionnelle est donc liée à la capacité de traiter avec le réel, capacité à l'origine de toute connaissance selon Aristote (1840), de dialoguer, de négocier. L'expérience en tant que contact avec le monde complexe dans lequel le sujet évolue, est donc fondamentale. Pour Vygotski, les expériences émotionnellement vécues appelées *perezhivanie* (перезивание, « au-delà » en russe), sont des zones

de développement des potentialités (Veresov, 2014). Elles sont considérées comme un processus de transformation indispensable au développement de la personne. L'identité personnelle et professionnelle se développe donc au sein d'une communauté grâce aux expériences médiées par les artefacts, le dialogue collaboratif et la sémiotique sociale (Longuet et Springer, 2021). Tout développement humain est « conjoint des autonomies individuelles, des participations communautaires et de la conscience d'appartenir à l'espèce humaine » (Morin, 2014, p. 105). L'expérience créative dialoguée ouvre la voie vers la connaissance de soi, de ses potentialités, du monde. En s'appropriant les artefacts, les enseignants s'énaquent (Maturana et Varela, 1994) et créent une culture partagée renouvelée.

## **Penser la complexité**

La programmation cocréative permet d'engager les enseignants dans un processus dialogique de recherche création pédagogique (Longuet, 2014). Elle contraint les enseignants à réaliser une production observable nouvelle adaptée à la situation (Bonnardel et Lubart, 2019). Elle crée ainsi les conditions de la recherche, de l'exploration, du dialogue qui cherche la compréhension des choses. Elle oblige les enseignants à se questionner, à identifier les situations problématiques inhérentes à la contrainte de programmation, à accepter le désordre, à affronter les incertitudes, à peser les contradictions et les complémentarités pour agir et arriver à une création acceptable. La création contraint les enseignants à « ne pas se contenter des modes d'emploi existants », à « se rejoindre dans une certaine abstraction prospective » (Rouquette, 1973, p. 122). Elle les oblige à développer une intelligence stratégique qui leur offre un certain nombre de scénarios possibles pour l'action qui peuvent être modifiés au cours du processus de création (Morin, 2014). La pensée stratégique relie de ce fait les enseignants ; elle les contraint à « langager », c'est-à-dire à se coordonner avec d'autres, à s'ajuster à d'autres pensées, à opérer dans le langage pour externaliser les problèmes, prendre des décisions, faire progresser la création, construire des connaissances (Longuet et Springer, 2021). La médiation est au cœur du processus dynamique de création. Le pôle du « langager » est central. C'est en quelque sorte le moteur de la pensée complexe. Il est essentiel pour agir. Il met en mouvement la pensée.

## **Développer des compétences dans une approche par situations**

L'Organisation des Nations unies pour l'éducation, la science et la culture (UNESCO) reconnaît que la prévalence des technologies de l'information et de la communication (TIC) est potentiellement

importante dans l'éducation et a développé le cadre de compétences TIC de l'UNESCO pour les enseignants (UNESCO, 2018), car il s'agit d'un outil utile pour guider la formation initiale et la recherche d'enseignants dans l'éducation. Avec ce référentiel, l'UNESCO vise à fournir une base pour l'élaboration de politiques actualisées et le renforcement des capacités dans le domaine dynamique des TIC. Le *Référentiel UNESCO de compétences TIC pour les enseignants* (2018) met en évidence le rôle que la technologie peut jouer dans le soutien de six domaines clés de l'éducation à travers trois phases d'apprentissage. Il est une référence pour les développements technologiques et pédagogiques récents dans les domaines des TIC et de l'éducation. Il souligne l'importance pour les enseignants de favoriser chez les élèves les compétences de collaboration, de résolution de problèmes et de créativité dans l'utilisation des technologies numériques (UNESCO, 2018).

À l'ère numérique, ces compétences deviennent partie intégrante de leur formation initiale (Montiel et Gómez-Zermeño, 2022). Elles peuvent être développées dans des situations de programmation cocreative. Pour cela, Wilson *et al.* (2012) proposent un cadre pour établir le niveau de compétence en programmation cocreative, en analysant un projet collaboratif en termes de concepts de programmation (tels que les fils d'exécution parallèles (*threads*), les instructions conditionnelles ou les variables), d'organisation du code (noms de variables, noms d'objets et blocs étranges), et de conception et de convivialité (telles que la fonctionnalité ou l'originalité, entre autres).

Les compétences documentées par l'UNESCO et les chercheurs sont dans notre cas construites, actualisées par les enseignants en situation de programmation cocreative. Elles émergent du processus de création. Jonnaert (2017, p. 44) considère qu'« une compétence se construit par des personnes en situations. Une compétence caractérise un moment particulier, celui de l'harmonie entre ces personnes et ces situations, le moment qui leur permet d'affirmer que leurs actions dans ces situations sont viables à cet instant. » Cette définition rappelle le caractère situé de la compétence, non prédéfini et l'importance de la relation entre les personnes. Les enseignants développent, complètent, transforment leur répertoire professionnel dans le cours de l'action dialoguée médiée par des artefacts matériels (logiciel de programmation éducative, robots éducatifs, etc.) et symboliques (langage de programmation, etc.). Leur curriculum se construit au fur et à mesure de la cocreation. La démarche choisie s'écarte de la démarche classique de la preuve étayée par le référentiel institutionnel. Les enseignants enrichissent leur répertoire professionnel didactique et pédagogique dans le cours de l'action et traduisent leur capital compétences dans un projet développé pour leurs élèves.

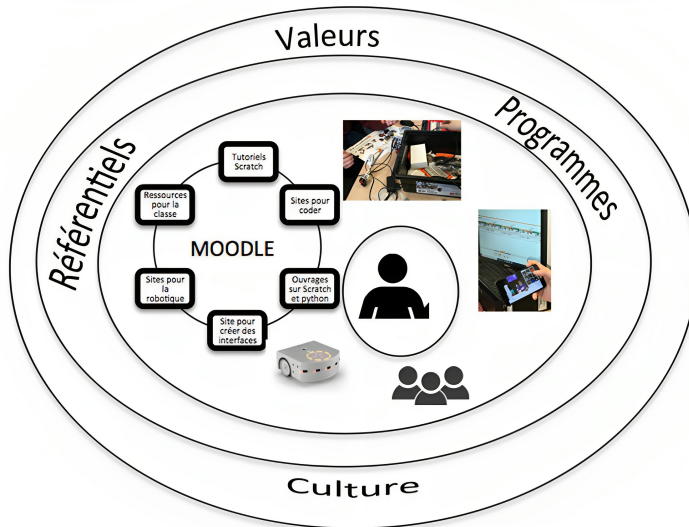


FIGURE 1 – Écosystème d’apprentissage et de recherche développé par le formateur INSPE.

## Dispositif de formation

### Un écosystème d’apprentissage et de recherche hybride

Le dispositif de formation est conçu à partir du modèle de Bronfenbrenner (1979). Il est écologique et sociosémiotique. Il favorise la reliance, c’est-à-dire la nécessité de se relier à soi et aux autres pour saisir des différences, des valeurs particulières, une compréhension du monde. Les enseignants évoluent dans un système environnemental complexe, systémique allant d’un microsystème à un macrosystème dans lequel chaque système communique avec un système plus vaste, renforçant ainsi la relation entre l’individu et son environnement comme le montre le schéma de la figure 1.

Les étudiants de l’INSPE disposent de la plateforme Moodle sur laquelle des ressources ont été déposées pour éclairer la contrainte de programmation cocréative. Le formateur a créé des liens référencés dans le tableau 1, vers des tutoriels pour apprendre à utiliser Scratch, des sites pour apprendre à coder comme Swift Playgrounds ou Algoblocs, des sites pour la robotique comme Thymio cycle 1.2.3, Lego ou Drones et robots à l’école, des ouvrages sur Scratch et Python et des ressources pour la classe comme Blockly games, Scratch exploreur studio.code.org/courses et Python lycée.

La plateforme Moodle est utilisée par les étudiants en présentiel et à distance. Le formateur a également mis à disposition



TABLEAU 1 – Liens référencés par le formateur.

Environnement et développement	<a href="https://scratch.mit.edu/ideas">https://scratch.mit.edu/ideas</a> <a href="http://www.scratchjr.org/">http://www.scratchjr.org/</a> <a href="https://www.apple.com/swift/playgrounds/">https://www.apple.com/swift/playgrounds/</a> <a href="https://www.algoblocs.fr/">https://www.algoblocs.fr/</a> <a href="https://repl.it/languages/python3">https://repl.it/languages/python3</a>
Ressources pour la classe	<a href="http://www.reseau-canope.fr/atelier-yvelines/spip.php?article1161">http://www.reseau-canope.fr/atelier-yvelines/spip.php?article1161</a> <a href="https://blockly.games/">https://blockly.games/</a> <a href="https://www.pedagogie.ac-aix-marseille.fr/jcms/c_-10544187/fr/scratch-explorateur">https://www.pedagogie.ac-aix-marseille.fr/jcms/c_-10544187/fr/scratch-explorateur</a> <a href="http://www.kidscod.in/">http://www.kidscod.in/</a> <a href="https://studio.code.org/courses">https://studio.code.org/courses</a> <a href="http://python.lycee.free.fr/">http://python.lycee.free.fr/</a>
Ouvrages sur Scratch, Python	<a href="https://www.eyrolles.com/Recherche/?q=scratch&amp;Rayon[INF]">https://www.eyrolles.com/Recherche/?q=scratch&amp;Rayon[INF]</a> <a href="https://www.eyrolles.com/Recherche/?q=python+kids">https://www.eyrolles.com/Recherche/?q=python+kids</a>
Robotique	<a href="https://www.terrapiologo.com/products/robots/bee-bot/bee-bot.html">https://www.terrapiologo.com/products/robots/bee-bot/bee-bot.html</a> cycle 1 <a href="https://www.thymio.org/fr/">https://www.thymio.org/fr/</a> cycle 2 1 3 <a href="https://www.lego.com/en-us/themes/mindstorms">https://www.lego.com/en-us/themes/mindstorms</a> cycle 4 <a href="https://paper.li/RfrentNumrique/1454506123#/">https://paper.li/RfrentNumrique/1454506123#/</a> drones et robots
Création d'interfaces	<a href="https://makeymakey.com/">https://makeymakey.com/</a>

des enseignants des kits Lego de construction de robots ainsi que des robots BeeBot. Cet environnement est augmenté par les étudiants s'ils le souhaitent. Certains mobilisent en présentiel pendant l'expérience cocreative des smartphones et des ordinateurs portables.

L'expérience de programmation cocreative s'inscrit dans une dynamique perception-action dans la mesure où les enseignants explorent l'environnement, sélectionnent ce qui est essentiel pour leur développement, voire l'enrichissent. Ils créent des partitions multiples, des *umwelts* (« environnements » en allemand; Von Uexküll, 2010) à chaque moment expérimental. Le formateur n'intervient pas dans le processus de cocreation. Il présente seulement l'écosystème d'apprentissage et de recherche lors de la première séance et il participe à la coévaluation finale en fin de TD.

## Présentation du public

La formation à la programmation cocreative s'est déroulée à l'INSPE Paris Sorbonne entre 2017 et 2019 dans le cadre de l'UE « projets pour la classe » du tronc commun structuré autour de quatre ensembles : les valeurs de la République, des gestes professionnels liés aux situations d'apprentissage et d'éducation, une

appropriation des thèmes transversaux d'éducation. Quarante-cinq étudiants du master MEEF, de l'éducation et de la formation, 1<sup>er</sup> et 2<sup>e</sup> degrés toutes disciplines confondues, ont suivi une formation d'une durée de 21 heures. L'objectif était d'initier des étudiants novices à l'informatique, à et par la programmation cocréative pour qu'ils construisent la compétence de programmation mais aussi pour qu'ils développent d'autres compétences en rapport avec le référentiel des enseignants comme la collaboration, la construction, la mise en œuvre et l'animation de situations d'enseignement et d'apprentissage prenant en compte la diversité des élèves.

### Démarche expérientielle : défi et contrainte de création

Le formateur a choisi de confronter les enseignants à trois défis puis à une contrainte de création pédagogique pour les engager dans un processus de création sans modèle de résolution les obligeant à explorer l'écosystème, à penser stratégiquement, à entrer dans une conversation réflexive schöniennne (Longuet, 2018) pour orienter leurs choix comme le montre la figure 2.

Les enseignants relèvent trois défis. Le premier consiste à créer un jeu permettant à des élèves de travailler les tables de multiplication :

Vous devez créer un jeu permettant à vos élèves de travailler les tables de multiplication. Un personnage pose les questions à vos élèves. L'élève commence avec cinq vies. Lorsque toutes les vies sont écoulées, un deuxième personnage vient manger le personnage principal. Les élèves doivent avoir des retours sur la validité ou non de leur réponse et le nombre de bonnes réponses.

Le second défi consiste à faire réaliser aux élèves un jeu permettant à un animal de se déplacer dans un labyrinthe pour atteindre une forme rouge en 45 secondes :

Le déplacement de l'animal se fait avec les flèches du clavier de l'ordinateur. À chaque fois que l'animal touche un bord du labyrinthe, il perd une vie. Au bout de cinq vies perdues, il revient au début. Une animation finale indique si le joueur a perdu ou gagné. Nombre de lignes de blocs : entre 35 et 40.

Le troisième défi consiste à réaliser un jeu sur les nombres pairs et impairs.



FIGURE 2 – Exploration de l'écosystème par les étudiants.

Vos élèves doivent réaliser un jeu sur les nombres pairs et impairs. Des nombres un à neuf descendent automatiquement et les élèves doivent cliquer sur ceux qui sont pairs (ou impairs). Progressivement les nombres descendent de plus en plus rapidement. Une vie est supprimée en cas d'erreur si le nombre n'a pas été cliqué en arrivant en bas. Les élèves ont un retour sur le nombre de vies et le nombre de bonnes réponses. Indice : il faut utiliser la fonction de clonage. Nombre de lignes de blocs : entre 27 et 30.

Ces défis les amènent à faire face à diverses situations de programmation cocréative en analysant un projet collaboratif en termes de concepts de programmation, de développement de leurs compétences de collaboration, de résolution de problèmes et de créativité dans le cours de l'action dialoguée médiée par des artefacts matériels et symboliques. Ils disposent ainsi d'un réservoir de solutions qu'ils pourront réutiliser ou explorer à nouveau, faire progresser.

Une fois les défis relevés, les enseignants conçoivent deux projets. Le premier consiste à concevoir un projet dans lequel les élèves développent un jeu avec Scratch. Le second consiste à concevoir un projet avec le robot Thymio ou Lego. Dans les deux cas, les enseignants stagiaires doivent remettre une fiche descriptive du projet, un exemple simulé du projet réalisé par un groupe d'élèves, la fiche de programmation sauvegardée depuis le logiciel, la photo d'un exemple simulé d'interface développée par les élèves et la vidéo de démonstration de l'activité réalisée avec le robot.

## Méthodologie de la recherche

### Une recherche située compréhensive

Cette étude s'inscrit dans une démarche de recherche située, compréhensive (Dayer et Charmillot, 2012). Elle vise à décrire et interpréter les dynamiques des participants. La réflexion inductive ancrée dans le paradigme d'exploration (Guillemette et Luckerhoff, 2012) permet une construction progressive du sens faisant apparaître des éléments qui en fin de compte font système. Ce type de recherche accorde une place significative au vécu des acteurs. Cela implique que l'on accepte l'idée que ces acteurs disposent d'une rationalité et d'une intentionnalité réelles et qu'ils ont la volonté de la partager et de dialoguer avec des partenaires (Longuet, 2018). Les traces, esquisses, photos, vidéos qui se déploient dans Moodle constituent le corpus de notre recherche. Elles dévoilent l'action stratégique, les prises de décision, le développement progressif d'acquis conceptuels.

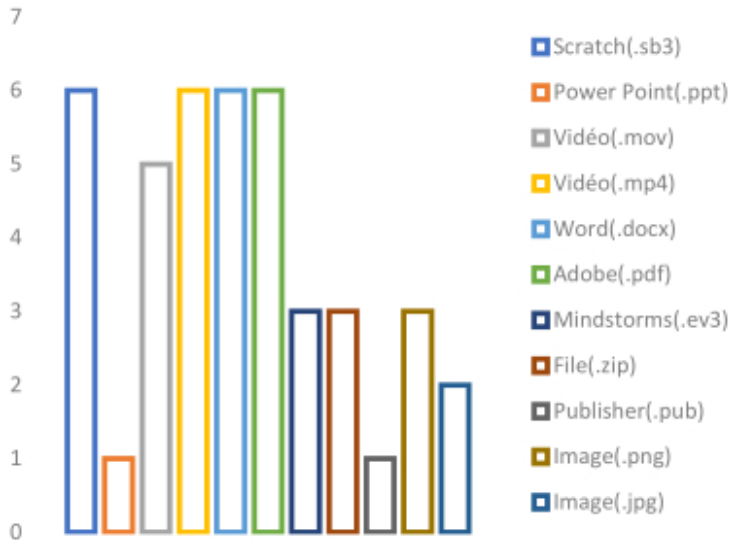


FIGURE 3 – Fichiers informatiques du portfolio électronique.

Nous considérons que cette démarche est pertinente, car il y a un déficit d’outils qui aident les enseignants à évaluer le développement de la pensée computationnelle (PC) et la correction des projets collectifs programmés (Montiel et Gómez-Zermeño, 2021; Wing, 2008). Concernant le langage de programmation Scratch, plusieurs auteurs ont proposé différentes approches pour évaluer le développement de la PC en analysant les projets des apprentis, mais la majorité de ces approches se basent exclusivement sur une analyse manuelle (Moreno *et al.*, 2015).

### Collecte des données

Brennan et Resnick (2012) ont introduit une stratégie basée sur l’analyse de portefeuilles de projets à l’aide d’un outil de visualisation appelé Scrape (Wolz *et al.*, 2011), complétée par des interviews sur les projets et la scénographie. Dans le but d’appliquer une démarche de recherche située dans la présente étude, les données générées par chacun des groupes de travail collaboratif ont été recueillies dans un portfolio électronique. La figure 3 montre à quels types de fichiers informatiques les données collectées correspondent.

### Analyse des traces, manifestations de l’activité cocréative des enseignants

Suivant les recommandations de Moreno *et al.* (2015), la démarche de recherche située a été appliquée à travers l’analyse des dossiers inscrits au portfolio de chacun des groupes d’enseignants qui ont réalisé un travail collectif sur la programmation informatique

cocréative en formation initiale. Pour cela, deux techniques d'analyse ont été appliquées, la première basée sur une méthode Delphi et la seconde appuyée par des outils d'analyse des codes Scratch programmés par chaque groupe d'enseignants stagiaires.

## Méthode Delphi

Conçue en 1950 par Olaf Helmer à la Rand Corporation (Okoli et Pawlowski, 2004), la méthode Delphi a pour but de mettre en évidence des convergences d'opinions et de dégager certains consensus sur des sujets précis, souvent avec un caractère prospectif important, grâce à la consultation d'experts. Aujourd'hui, la méthode Delphi est de plus en plus utilisée par plusieurs chercheurs en développement des modèles dans les domaines des systèmes d'information, de la gestion des connaissances et des technologies de l'information (TI), spécifiquement en développement des capacités en gestion de projets en TI.

Pour appliquer la méthode Delphi, une équipe d'experts a été constituée et composée de deux docteurs en innovation pédagogique, d'un docteur en éducation et TIC (*e-learning*) et deux ingénieurs en systèmes informatiques.

Dans le but d'analyser les fichiers que les enseignants ont enregistrés dans leurs portfolios, l'équipe d'experts a sélectionné un ensemble de compétences et sous-compétences du cadre de compétences en TIC proposé par l'UNESCO ; ils ont complété cette liste en s'appuyant sur les idées théoriques sur la pensée complexe, appliquées à la programmation cocréative.

Lors de l'analyse, les experts ont utilisé un guide d'observation composé de la liste des compétences et sous-compétences sélectionnées, pour évaluer les trois niveaux établis dans le but de générer des informations qui leur permettraient d'interpréter la dynamique générée par les groupes d'enseignants au cours de leur travail collaboratif de programmation cocréative.

Ces trois niveaux de compétences sont basés sur le cadre de compétences TIC pour les enseignants de l'UNESCO (UNESCO-ICTCFT) :

- Niveau 1 : acquisition de connaissances ;
- Niveau 2 : approfondissement des connaissances ;
- Niveau 3 : création de connaissances.

Le guide d'observation (tableau 2), a été développé avec les experts sur la base des idées théoriques des auteurs référencés dans cette étude. Les trois colonnes de droite de ce tableau correspondent aux trois niveaux explicités ci-dessus.

TABLEAU 2 – Guide d’observation des experts.

Compétences	Sous-compétences	Niveau 1	Niveau 2	Niveau 3
Programmation informatique	Traitement de l’information			
	Résolution de problèmes			
	Créativité			
	Concevoir des systèmes			
	Complexité			
	Pensée procédurale			
Programmation cocréative	Pensée informatique			
	Contrainte de création			
	Collaboration			
	Médiations			
	Expérience partagée			
	Langager			
Pensée complexe	Sémiotique sociale			
	Processus de cocréativité			
	Affronter les incertitudes			
	Peser les contradictions et complémentarités			
	Outils complexes			
	Écosystème d’apprentissage			
	Démarche expérientielle			
Pensée critique				
	Pensée stratégique			

## Dr. Scratch

Boe *et al.* (2013) ont développé Hairball, un analyseur de code statique qui détecte les problèmes potentiels dans les programmes, tels que le code qui ne s’exécute jamais, les messages qu’aucun objet ne reçoit ou les attributs qui ne sont pas correctement initialisés. Basé sur Hairball, Dr. Scratch est une application Web qui permet d’analyser automatiquement les projets Scratch pour vérifier s’ils ont été programmés correctement et recevoir des commentaires pour améliorer leur code, et développer leur capacité de PC.

Dr. Scratch détecte certaines mauvaises habitudes de programmation ou bogues potentiels, tels que l’utilisation de noms sans signification, le code redondant, le code qui ne s’exécute pas et l’initialisation incorrecte des attributs d’objet. Pour attribuer le score PC, Dr. Scratch déduit la capacité de réflexion computationnelle démontrée par le programmeur en fonction des sous-compétences (Moreno *et al.*, 2015).

1. Parallélisme : développer un programme qui exécute plusieurs actions en même temps.
2. Pensée logique : reconnaître le problème à résoudre, trouver une solution à programmer et analyser les résultats obtenus pour en tirer des conclusions.
3. Algorithmes de contrôle de flux : contrôlent les instructions ou les blocs qui effectuent les différentes actions.
4. Interactivité avec l'utilisateur : développer des programmes qui permettent aux utilisateurs de se sentir intégrés au jeu vidéo, à l'histoire ou au projet que nous créons, en travaillant d'une manière ou d'une autre.
5. Abstraction et décomposition des problèmes : développer des compétences de programmation complexes qui permettent de diviser un gros problème (ou un défi) en petits problèmes faciles à résoudre.
6. Représentation des informations : identifier l'ensemble des données et variables qui ont une valeur à chaque instant de l'exécution du programme, et qui peuvent être modifiées par les instructions ou la séquence de blocs.
7. Synchronisation : permet à un programme de suivre un certain ordre et d'effectuer une action lorsqu'un autre a terminé, formant une chaîne ordonnée d'actions.

Tous les programmes réalisés par les étudiants dans leur travail de cocréation collaborative ont été analysés avec Dr. Scratch. Il a été observé que la sous-compétence de l'interactivité avec les utilisateurs a obtenu les résultats les plus faibles, tandis que le contrôle de flux et l'abstraction ont obtenu les résultats les plus élevés. Ces résultats témoignent des compétences qu'ils ont développées pour résoudre des problèmes complexes et l'utilisation d'outils complexes.

De leur côté, les experts ont passé en revue les portfolios et les codes de chacun des programmes Scratch développés par les groupes d'étudiants. Ainsi, ils ont rempli un guide d'observation (tableau 2) pour chaque groupe, indiquant le niveau de sous-compétence qu'ils ont considéré, en fonction de leur expérience en programmation informatique cocreative et de leur connaissance de Scratch, lors de l'évaluation de chaque portfolio.

Les résultats montrent que les enseignants stagiaires novices en programmation ont développé des compétences en programmation informatique grâce à la programmation Scratch. Ils ont néanmoins atteint un stade de développement plus important en créativité et en pensée complexe. La programmation informatique a participé au développement de la pensée complexe et de la créativité.

L'analyse des portfolios a également révélé des imaginaires pluriels bien que marqués par les référentiels et les programmes en fonction

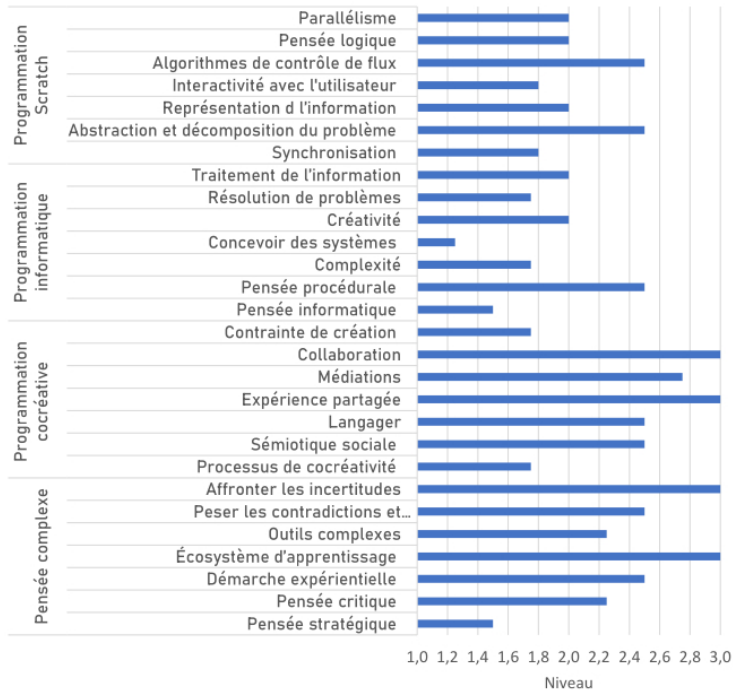


FIGURE 4 – Résultats de l'analyse avec la méthode Delphi et avec l'outil Dr. Scratch.

des milieux traversés par les enseignants stagiaires, de leur héritage culturel et de leurs valeurs. Elle a dévoilé le processus de stylisation professionnelle (Clot et Faïta, 2000) nécessaire au développement professionnel.

L'analyse confirme l'influence de l'écosystème sur le développement des personnes. Certains groupes ont perçu la programmation cocreative comme un moyen de transfert des apprentissages disciplinaires et de vérification des acquis. Le projet a servi par exemple à vérifier des acquis en histoire en cycle 3 (développement d'une conscience historique) ou en français au collège (structure du conte). D'autres groupes ont pensé la programmation cocreative comme un moyen d'apprentissage, un moyen par exemple de passer de l'espace vécu à l'espace représenté. D'autres groupes ont imaginé la programmation cocreative comme un levier pour vivre ensemble, inclure. Certains projets ont été développés pour favoriser la liaison CM2/6<sup>e</sup>. Enfin, plusieurs groupes ont considéré la programmation comme un jeu. Les enseignants stagiaires ont exploré un environnement et sélectionné ce qui leur a semblé nécessaire pour leur développement professionnel. Chaque groupe a joué sa partition comme l'explique Uexküll (2010) à travers sa métaphore musicale. Chaque groupe constitué de compositeurs/programmeurs a orchestré un morceau particulier, tissé des liens pour se développer personnellement et professionnellement, s'inscrire dans un projet sociétal d'éducation.



## Discussion et conclusion

Notre recherche s'est intéressée à une formation innovante à et par la programmation cocréative en rupture avec les formations transmissives (Romero *et al.*, 2017) et individuelles. La formation observée fondée sur l'expérience partagée et la conception d'un écosystème d'apprentissage et de recherche ouvert a embrassé la complexité pour favoriser l'émergence des zones de potentialité de chaque enseignant stagiaire.

Les résultats montrent que la stylisation professionnelle liée à l'action créatrice partagée a un impact important sur le développement de la compétence de programmation et les compétences du référentiel enseignant. Il semble de ce fait intéressant de considérer les enseignants stagiaires comme des acteurs créatifs curieux disposant d'un répertoire pluriel, de favoriser les opérations complexes pour qu'ils s'approprient les genres professionnels et les transforment, qu'ils puissent se projeter dans le monde professionnel dans lequel ils évoluent plutôt que d'être réduits à imiter les genres. Le passage à une pédagogie fondée sur la reliance, l'écologie, l'innovation nécessite néanmoins une transformation profonde des postures professionnelles des formateurs et des enseignants. La salle de formation et de classe doit se transformer en un véritable laboratoire social dans lequel les différentes partitions peuvent s'exprimer. Par ailleurs, les résultats de cette recherche mettent en évidence la nécessité d'intégrer des stratégies de développement des compétences en programmation informatique cocréative dans la formation initiale étant donné qu'elles permettent le développement de compétences qui renforcent la pensée complexe. Pour les recherches futures, il est recommandé de poursuivre l'analyse des stratégies pour impliquer les enseignants débutants dans l'utilisation d'outils qui vont au-delà de la programmation procédurale, dans le but de développer une pensée complexe. Une approche par situations de programmation cocréative de plus en plus complexes en formation initiale des enseignants pourrait être une piste, la compétence se situant dans un continuum.

En effet, la programmation cocréative est un outil fondamental pour faire face aux défis de l'éducation en termes de complexité, de diversité, d'interdisciplinarité et de créativité, qui permet de résoudre des problèmes complexes par la pensée systémique. Comprendre la complexité de la réalité vécue par les enseignants débutants dans les écoles, souscrit à la fois à une vision philosophique existentialiste qui considère la construction de l'identité personnelle et professionnelle comme un atout, et à une anthropologie qui place l'être humain au centre de sa formation.

Cette étude a été réalisée dans le cadre de la collaboration offerte par le projet pluridisciplinaire « Informatique à l'école : conceptualisations, accompagnement, ressources » (IE CARE), qui a

permis l'association de chercheurs internationaux reconnus pour leurs contributions dans les domaines des sciences humaines, des sciences sociales et des TIC. Leurs résultats corroborent d'une part avec la nécessité d'identifier les convergences des travaux de recherche dans le domaine des sciences humaines, sociales et des TIC ; comprendre, évaluer et valider des scénarios pédagogiques et des ressources pédagogiques numériques ; et d'autre part de construire un cadre de stratégies et d'indicateurs dans le but de développer des cultures numériques pour la formation des enseignants à l'innovation pédagogique.

### Recommandations

Pour instaurer une formation à et par la programmation cocréative, il paraît nécessaire de passer d'une pédagogie centrée sur l'objet programmation à une pédagogie de la relation, écologique et sociosémiotique basée sur le partage des connaissances, la confrontation des approches, la multiplication des regards. Pour ce faire, il convient d'engager les apprenants au sein d'écosystèmes d'apprentissage et de recherche favorisant la reliance et le dialogue collaboratif (langager), notamment à travers l'imposition de contraintes créatives.

### Références

- Alemán de la Garza, L., Bruillard, E., Gómez-Zermeño, M., *et al.* (2019). *Rethinking pedagogy : Exploring the potential of digital technology in achieving quality education*, UNESCO-UNESDOC Mahatma Gandhi Institute of Education for Peace and Sustainable Development (MGIEP). <https://unesdoc.unesco.org/ark:/48223/pf0000372786>
- Aristote (1840). *Métaphysique, Livre I.* (Victor Cousin trad., texte originel publié au IV<sup>e</sup> siècle av. J.-C.). <http://remacle.org/bloodwolf/philosophes/Aristote/metaphysique1fr.htm>
- Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., et Franklin, D. (2013). Hairball : Lint-inspired static analysis of scratch projects, *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013)*, p. 215-220. <https://doi.org/10.1145/2445196.2445265>
- Bonnardel, N., et Lubart, T. (2019). La créativité : approches et méthodes en psychologie et en ergonomie, *RIMHE : Revue interdisciplinaire management, homme et entreprise*, vol. 37, n° 8, p. 79-98. <https://doi.org/10.3917/rimhe.037.0079>
- Brennan, K., et Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research*

Association (AERA 2012), vol. 1. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>

Bronfenbrenner, U. (1979). *The Ecology of Human Development : Experiments by Nature and Design*, Harvard University Press. <https://doi.org/10.4159/9780674028845>

Bruillard, É., et Fluckiger, C. (2018). Une approche didactique de l'informatique scolaire, *Revue française de pédagogie, recherche en éducation*, vol. 204, p. 86-87. <https://doi.org/10.4000/rfp.8523>

Clot, Y., et Faïta, D. (2000). Genres et styles en analyse de travail. Concepts et méthodes, *Travailler*, vol. 4, p. 7-42.

Dayer, C., et M. Charmillot (2012). Démarche compréhensive et méthodes qualitatives : clarifications épistémologiques, *Formations et pratiques d'enseignement en questions*, vol. 14, p. 163-176

Dorais, L.-J. (2004). « La construction de l'identité », dans Denise Deshaies et Diane Vincent (dir.), *Discours et constructions identitaires*, Presses de l'université de Laval.

Gosselin, P., et Le Coguiec, E. (2006). *La recherche création, Pour une compréhension de la recherche en pratique artistique*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18ph3x1>

Guillemette, F., et Luckerhoff, J. (2012). *L'induction en méthodologie de la théorisation enracinée (MTE). Fondements, procédures et usages*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18pgxtm>

Jonnaert, P. (2017). La notion de compétence : une réflexion toujours inachevée, *Éthique publique*, vol. 19 n° 1. <https://doi.org/10.4000/ethiquepublique.2932>

Longuet, F. (2012). *L'impact des outils d'évaluation qualitative et du WEB 2.0 sur le développement et l'identification des compétences professionnelles des enseignants de langues*, thèse de l'université Sorbonne Nouvelle - Paris III. <http://tel.archives-ouvertes.fr/tel-00770640/>

Longuet, F. (2014). Former les enseignants de Français langue étrangère par l'activité de création numérique dialoguée, *Synergies Espagne*, vol. 7, p. 189-204. <http://gerflint.fr/Base/Espagne7/Longuet.pdf>

Longuet, F. (2018). Créativité, dialogisme et multimodalité : trois qualités d'un environnement de conception numérique collaboratif, *Revue internationale de pédagogie de l'enseignement supérieur*, vol. 34, n° 2. <https://doi.org/10.4000/ripes.1450>

Longuet, F., et Springer, C. (2021). *Autour du CECR Volume Complémentaire (2018) : Médiation et collaboration. Une didactique de la relation, écologique et sociosémiotique*, Éditions des archives contemporaines. <https://doi.org/10.17184/eac.9782813004055>

Maturana, H., et Varela, F. (1994). *L'arbre de la connaissance*, Addison Wesley France.

Montiel, H., et Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in k-12 education : A systematic literature review of “scratch” as an innovative programming tool, *Computers*, vol. 10, n° 6, p. 69. <https://doi.org/10.3390/computers10060069>

Montiel, H., et Gomez-Zermeño, M. G. (2022). Rock the Boat! Shaken by the COVID-19 Crisis : A Review on Teachers' Competencies in ICT, *Frontiers in Education*, vol. 6, p. 558. <https://doi.org/10.3389/educ.2021.770442>

Moreno-León, J., Robles, G., et Román-González, M. (2015). Dr. Scratch : análisis automático de proyectos Scratch para evaluar y fomentar el pensamiento computacional, *Revista de Educación a Distancia (RED)*, vol. 46, n° 10. <https://doi.org/10.6018/red/46/10>

Morin, E. (2014). *Introduction à la pensée complexe*, Éditions Points.

Okoli, C., et Pawlowski, S. D. (2004). The Delphi method as a research tool : an example, design considerations and applications, *Information et Management*, vol. 42, p. 15-29. <https://doi.org/10.1016/j.im.2003.11.002>

Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative, *Formation et Profession*, vol. 24, n° 1, p. 87-89. <https://doi.org/10.18162/fp.2016.a92>

Romero, M., Lepage, A., et Lille, B. (2017). Computational thinking development through creative programming in higher education, *International Journal of Educational Technology in Higher Education*, vol. 14, n° 1, p. 1-15. <https://doi.org/10.1186/s41239-017-0080-z>

Rouquette, M.-L. (1973). *La créativité*, Presses universitaires de France.

UNESCO (2018). *Référentiel UNESCO de compétences TIC pour les enseignants*, UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000368966>

Veresov, N. (2014). « Émotions, Perekhivanie et développement culturel : le projet inachevé de Lev vygotski », dans C. Moro et N. Muller Mirza (dir.), *Sémiotique, Culture et Développement Psychologique*, Presses universitaires du Septentrion.

Von Uexküll, J. (2010). *Milieu animal et milieu humain*, Bibliothèques Rivages.

Voogt, J., et Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences : Implications for national curriculum policies, *Journal of curriculum studies*, vol. 44, n° 3, p. 299-321. <https://doi.org/10.1080/00220272.2012.668938>

Wilson, A., Hainey, T., et Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts, *Proceedings of the 6th European Conference on Games-Based Learning (ECGBL)*, p. 4-5.

Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, vol. 366, n° 1881, p. 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>

Wolz, U., Hallberg, C., et Taylor, B. (2011). Scrape : A tool for visualizing the code of scratch programs, *Poster présenté à la 42<sup>e</sup> ACM Technical Symposium on Computer Science Education (SIGCSE 2011)*, Dallas, États-Unis.

Pour citer ce chapitre :

Longuet, Frédérique, et Gómez-Zermeño, Marcela Georgina (2024). « Former les enseignants à et par la programmation informatique cocréative en formation initiale », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 193-210. <https://doi.org/10.53480/2024iecare10x>



# Assister les enseignants du primaire dans l'enseignement de la pensée informatique. Une approche basée sur la scénarisation et vers les *teaching analytics*

# 11

Amel YESSAD<sup>1</sup>  
Mathieu MURATET<sup>2</sup>  
Thibault CARRON<sup>1</sup>

1. Sorbonne Université, CNRS, LIP6,  
75007 PARIS, France

2. INSEL, 92150 Suresnes, France

Dans un contexte où plusieurs pays ont introduit la science et la pensée informatique (PI) dans leurs curricula, où le codage a fait son entrée dans les programmes français à partir de 2016 (Bulletin officiel spécial n° 11 du 26 novembre 2015) et où les initiatives se multiplient en faveur d'un enseignement de la science informatique en France dès le plus jeune âge (Class'Code<sup>1</sup>, 1,2,3 Codez!<sup>2</sup>), il devient important d'explicitier les objectifs d'un tel enseignement, son interaction avec d'autres disciplines (en France, l'enseignement de l'informatique est souvent abordé à travers et pour l'enseignement des mathématiques) et d'organiser la formation des enseignants. Nous pensons que plusieurs leviers sont possibles pour préparer les enseignants à l'enseignement de la PI et *a minima* soutenir leur activité en classe, allant de la formation initiale ou continue classique à la proposition d'outils numériques d'aide à la préparation des séances. Dans cette perspective, les outils numériques peuvent-ils soutenir l'activité professionnelle des enseignants de primaire et jouer le rôle de facilitateurs leur permettant ainsi de mieux préparer leurs séances d'introduction à la PI? La réponse à cette question est complexe pour plusieurs raisons. D'une part, l'utilisabilité de ces outils peut représenter un frein à leur usage effectif par des enseignants qui ne sont pas toujours familiers avec le numérique (Holden et Rada, 2011; Storey *et al.*, 2002) et d'autre part, l'utilité et l'impact que pourraient avoir ces outils sur la pratique des enseignants à moyen et long termes ne sont pas suffisamment montrés dans l'état de l'art (Brinkerhoff, 2006; Watson, 2006).

Toutefois, de nombreuses ressources pour l'introduction de la PI à l'école primaire sont disponibles. Certaines mettent en avant l'informatique débranchée qui est un point d'entrée de la science informatique et qui présente l'avantage de ne pas nécessiter de matériels coûteux (Alayrangues *et al.*, 2017; Romero *et al.*, 2018). D'autres ressources exploitent des robots programmables où la

1. <https://pixees.fr/classcode-v2/>,  
accédé le 12/10/2023

2. <https://fondation-lamap.org/projet/123-codez>,  
accédé le 12/10/2023

nature tangible de ces objets fournit un artefact fertile sur le plan cognitif pour développer des compétences mathématiques et des formes de pensées algorithmiques (Komis et Misirli, 2011). Ces robots programmables sont alors exploités dans le cadre de scénarios pédagogiques. Enfin, de nombreuses ressources numériques et ludiques complètent ce panorama (Miljanovic et Bradbury, 2018; Vahldick *et al.*, 2014). Ces environnements informatiques pour l'apprentissage humain (EIAH) couvrent un spectre large allant de l'enseignement primaire à l'université; de l'initiation à la programmation avec des langages à base de blocs à la programmation d'intelligences artificielles avec des langages avancés. L'offre est donc riche et consistante même pour des enseignements à destination de l'école primaire. Pour autant l'appropriation de ces ressources par les enseignants reste un enjeu. Alayrangues *et al.* (2017) notent que mener des activités débranchées requiert des connaissances en informatique et des compétences en pédagogie; Spach (2019) souligne, dans le contexte de la robotique pédagogique, les freins au développement de ces ressources : « le défaut de maîtrise conceptuelle des enseignants est sans doute à l'origine du manque de référencement et du déficit d'institutionnalisation des notions et des concepts abordés dans les situations pédagogiques ». Concernant les EIAH dédiés à l'apprentissage de la PI, ils restent difficilement adaptables (Saddoug *et al.*, 2022) car d'une part la plupart de ces EIAH ne fournissent pas d'outils informatiques simples permettant de modifier les scénarisations existantes ou d'en créer de nouvelles et, d'autre part, pour les rares qui le permettent comme Kodu<sup>3</sup>, Scratch<sup>4</sup> ou Code.org<sup>5</sup>, ils ne proposent pas d'aide à l'enseignant pour comprendre les situations créées comme par exemple expliciter les compétences en jeu dans ces situations.

3. Kodu : <https://www.kodugamelab.com/>, accédé le 12/10/2023

4. Scratch : <https://scratch.mit.edu/>, accédé le 12/10/2023

5. Code.org : <https://code.org/>, accédé le 12/10/2023

Nos recherches dans le cadre du projet « Informatique à l'école : conceptualisations, accompagnement, ressources » (IE CARE) de l'Agence nationale de la recherche (ANR) s'intéressent à la définition, au développement et à l'évaluation d'outils d'aide pour assister les enseignants lors de leurs séances d'introduction de la PI. Notre problématique de recherche concerne les moyens à mettre en œuvre pour aider les enseignants de primaire à enseigner les concepts de la PI à leurs jeunes élèves. Quel serait l'apport de la recherche de manière générale et celui de la recherche en EIAH pour réduire le fossé qui existe entre les capacités des enseignants du primaire et les attendus de l'institution, et plus globalement de notre monde de plus en plus numérisé, connecté et régi par des systèmes intelligents afin de leur permettre de préparer au mieux les élèves aux défis de demain ?

Dans le cadre de ce chapitre, nous présentons une recherche qui a pour objectif de tirer profit du domaine des *teaching analytics* pour comprendre les pratiques des enseignants lors du processus de scénarisation de leurs séances. Cette recherche est

un travail préliminaire faisant partie d'une recherche longitudinale et pluridisciplinaire visant à apporter des réponses à notre problématique.

Ainsi, nous axons ce chapitre sur une contribution qui a consisté à développer avec des enseignants de primaire une plateforme de scénarisation de ressources pédagogiques sur le thème de la PI : cet outil d'aide à la scénarisation s'appuie sur un modèle conceptuel et a fait l'objet d'une conception centrée utilisateur, de développement itératif et d'études à la fois qualitative et quantitative. Nous analysons les usages des enseignants vis-à-vis de cet outil dans la perspective d'enrichir la plateforme avec un système de rétroactions.

## Revue de la littérature

### Pensée informatique et pratiques des enseignants

Le concept de « pensée informatique » a été employé pour la première fois par Papert (1993) pour mettre en avant le pouvoir des machines à améliorer la pensée des enfants et leurs schémas (*patterns*) d'acquisition de connaissances. Toutefois, l'article publié par Denning (2017) fait le constat que les concepts qui sous-tendent la PI sont beaucoup plus anciens. Par exemple, dès 1945, George Polya écrivait sur les disciplines et les méthodes mentales permettant de résoudre des problèmes mathématiques (Pólya et Conway, 1957). Son livre *How to Solve It* constituait déjà une sorte de précurseur de la PI. Aussi, en 1960, Alan Perlis affirmait que le concept d'*algorithmizing* faisait partie de notre culture. Il soutenait que les ordinateurs automatiseraient les tâches humaines et que l'algorithmique finirait par apparaître dans tous les domaines (Denning, 2017).

Plus tard, c'est Jeannette Wing qui a popularisé le terme de PI en mettant en avant son intérêt pour l'enseignement des sciences, des technologies, de l'ingénierie et des mathématiques (Wing, 2006). Elle la présente comme un savoir fondamental du XXI<sup>e</sup> siècle, qui devrait faire partie des apprentissages de l'école primaire. L'Académie des sciences va dans ce sens et recommande en 2013 : « La décision essentielle à prendre est de mettre en place un enseignement de science informatique depuis le primaire jusqu'au lycée, orienté vers la compréhension et la maîtrise de l'informatique, et dépassant donc largement les seuls usages des matériels et logiciels. Cette mise en place ne doit plus être différée » (Abiteboul *et al.*, 2015).



La définition de la PI est une question vive (Brennan et Resnick, 2012 ; Lee *et al.*, 2011 ; Selby et Woollard, 2013), plusieurs chercheurs la définissent comme un ensemble de processus cognitifs mobilisés lors de la résolution de problèmes, allant de la définition du problème, sa modélisation en faisant abstraction des éléments non utiles pour le résoudre, la recherche d'une solution généralisable à la mise en œuvre de cette solution de manière optimale. À l'instar de la démarche d'un informaticien, les problèmes complexes sont alors décomposés en problèmes plus simples à résoudre (Selby et Woollard, 2013 ; Wing, 2006).

Dès 2016, l'informatique a fait son entrée dans les programmes de l'école primaire sous la forme d'une sensibilisation à la programmation. Les textes préconisent explicitement la mise en œuvre d'activités de codage pour déplacer un robot ou un personnage à l'écran dans une approche intégrée aux autres apprentissages, en particulier les mathématiques (repérage et déplacements dans l'espace ou construction de figures géométriques). Sans faire référence à la notion de PI, les compétences visées restent proches<sup>6</sup> : (1) se repérer, s'orienter en utilisant des repères ; (2) adopter une démarche scientifique : utilisation d'un langage spécifique, contrôle, essais-erreurs ; (3) développer l'abstraction : apprendre à anticiper l'effet de telle ou telle séquence d'instructions avant même de la faire exécuter par une machine ou un programme. Parmentier (2018) propose un découpage fin des compétences de la pensée informatique et algorithmique pour l'enseignement fondamental (PIAF).

Plusieurs travaux de recherche (Alayrangues *et al.*, 2017 ; Kradolfer *et al.*, 2014 ; Spach, 2019) ont mis en avant les difficultés rencontrées par les enseignants du primaire pour prendre en main ces ressources qui s'expliquent en grande partie par leur manque de formation. Pour autant des ressources de formation existent, la page d'accueil d'Eduscol consacrée au numérique dans le premier degré propose un parcours M@gistere (premières activités de programmation pour comprendre le numérique) élaboré par des acteurs extérieurs à l'Éducation nationale (Class'Code portée par l'Institut national de recherche en sciences et technologies du numérique – INRIA, et D-Clics numériques portée par la Ligue de l'enseignement) et qui vise à transmettre les fondements historiques de la PI en préambule à des activités d'introduction à la programmation. Des initiatives existent également au sein de l'éducation nationale dans le cadre de formations continues et initiales (Parmentier, 2018). Malgré toutes ces initiatives, le passage à l'échelle de l'enseignement de la PI reste difficile, en particulier auprès des enseignants non formés à celle-ci.

6. Dossier thématique d'eduscol sur l'initiation à la programmation en cycle 2 et 3 : <https://eduscol.education.fr/document/15409/download>, accédé le 12/10/2023

## ***Scenario-based learning***

Dans cette recherche, nous nous appuyons sur le cadre théorique de l'apprentissage basé sur des scénarios ou *scenario-based learning* (SBL) (Schrader *et al.*, 2004; Seren Smith *et al.*, 2018) qui considère les scénarios comme un vecteur important dans les processus d'apprentissage des élèves et celui des enseignants, en particulier les débutants. Dans le contexte de l'approche SBL, un scénario consiste à mettre en situation des élèves à travers des ressources de types simulations, résolution de problèmes ou projets concrets. On a montré dans plusieurs recherches que les scénarios donnent la possibilité aux apprenants d'expérimenter efficacement leurs connaissances, de connecter la théorie à la pratique (De Coninck *et al.*, 2019), de développer leur esprit critique en explorant différentes possibilités et différentes méthodes de résolution de problèmes (Seren Smith *et al.*, 2018). Des recherches ont aussi montré l'impact positif de cette approche sur des enseignants débutants leur permettant d'aborder plus simplement et plus concrètement des connaissances pouvant être difficiles à transmettre (Hursen et Fasli, 2017). On a également montré que l'approche SBL favorise la rétention des connaissances sur le long terme (Brown *et al.*, 2015) et impacte positivement, *a fortiori* lorsqu'elle est combinée à des rétroactions et de la réflexion, le sentiment d'auto-efficacité et de bonne préparation des enseignants débutants avant la classe (Anseel *et al.*, 2009; Bardach *et al.*, 2021; Klassen *et al.*, 2021). Dans ce cadre, nous avons développé la plateforme de scénarisation ScenoClasse qui a pour objectif d'aider les enseignants de primaire, peu expérimentés et peu formés à la PI, à préparer leurs séances d'introduction. ScenoClasse permet d'orchestrer des ressources différentes et très variées autour de la PI et est ancrée dans le cadre théorique de l'approche SBL.

## ***Teaching analytics* ou l'analyse des comportements enseignants**

Les *teaching analytics* sont définis comme la mesure, le recueil, l'analyse et la synthèse de traces laissées par des enseignants pour comprendre leurs comportements et décisions pédagogiques dans le but d'optimiser leur impact sur les apprentissages des élèves. Dès 2012, on s'est intéressé à analyser les traces laissées par les enseignants sur différentes plateformes d'enseignement ou de préparation de séances de cours (Goggins, 2012). Une revue de la littérature à ce sujet peut être trouvée chez Bennacer (2022); Sergis et Sampson (2017) ou encore dans Ndukwe et Daniel (2020) où un panorama exhaustif sur les approches proposées, notamment pour le choix des indicateurs et les types de visualisations est dressé. Concernant la pratique de l'enseignant lui-même, les travaux de

Ku et ses collègues (Ku *et al.*, 2018) et de Chen (2020) ont montré, avec une approche *teaching analytics* et notamment l'utilisation d'analyses visuelles de l'apprentissage pour Chen, un impact sur la conception et la préparation des cours mais également la manière de les dispenser.

Dans le cadre de ce travail, l'analyse des traces numériques de la plateforme de scénarisation nous a permis d'identifier et d'extraire des comportements d'usages des enseignants. Ces comportements d'usages pourraient être utiles par exemple pour afficher des rétroactions adaptées aux enseignants utilisant la plateforme ScenoClasse. Ce système de rétroactions n'est pas l'objet de ce chapitre mais une des perspectives de ce travail.

## Présentation de la plateforme de scénarisation ScenoClasse et du modèle sous-jacent

Un modèle de description de tâches et d'organisation de tâches en scénarios a été conçu, lors de deux ateliers, avec 22 enseignants de primaire en formation (master métiers de l'enseignement, de l'éducation et de la formation – MEEF – mention premier degré) et huit enseignants expérimentés (maternelle/élémentaire) n'ayant jamais mené d'activité d'apprentissage de l'informatique ou de la PI. Durant ces deux ateliers, nous avons suivi une démarche de conception centrée utilisateurs (Brunet *et al.*, 2020). Partant de l'hypothèse que l'activité de scénarisation pourrait soutenir la pratique professionnelle des enseignants débutants, les entretiens menés avec eux ont permis d'identifier certains besoins dans les scénarios. Par exemple, le besoin d'explicitier les compétences travaillées dans le scénario (PIAF, cadre de référence des compétences numériques – CRCN<sup>7</sup>, etc.), de permettre une organisation spatio-temporelle flexible du scénario, de proposer différents niveaux de granularité (tâche ou activité, scénario, séquence de scénarios, etc.) ou de rendre adaptable le scénario à l'hétérogénéité des classes et des élèves au sein d'une même classe. Les éléments de réponse apportés, relatifs à la granularité du scénario, au langage de description et à la formalisation des scénarios et leur propriété d'adaptation, ont été inspirés par les modèles existants et l'état de l'art, en particulier les travaux de Dillenbourg sur les graphes d'orchestration (Dillenbourg, 2015) représentant le flux d'exécution des activités par les élèves individuellement, en groupe ou en classe entière, les travaux sur la théorie de l'activité et ceux de Komis sur la formalisation de la scénarisation (Komis *et al.*, 2013).

7. CRCN : <https://eduscol.education.fr/721/evaluer-et-certifier-les-competences-numeriques>, accédé le 12/10/2023

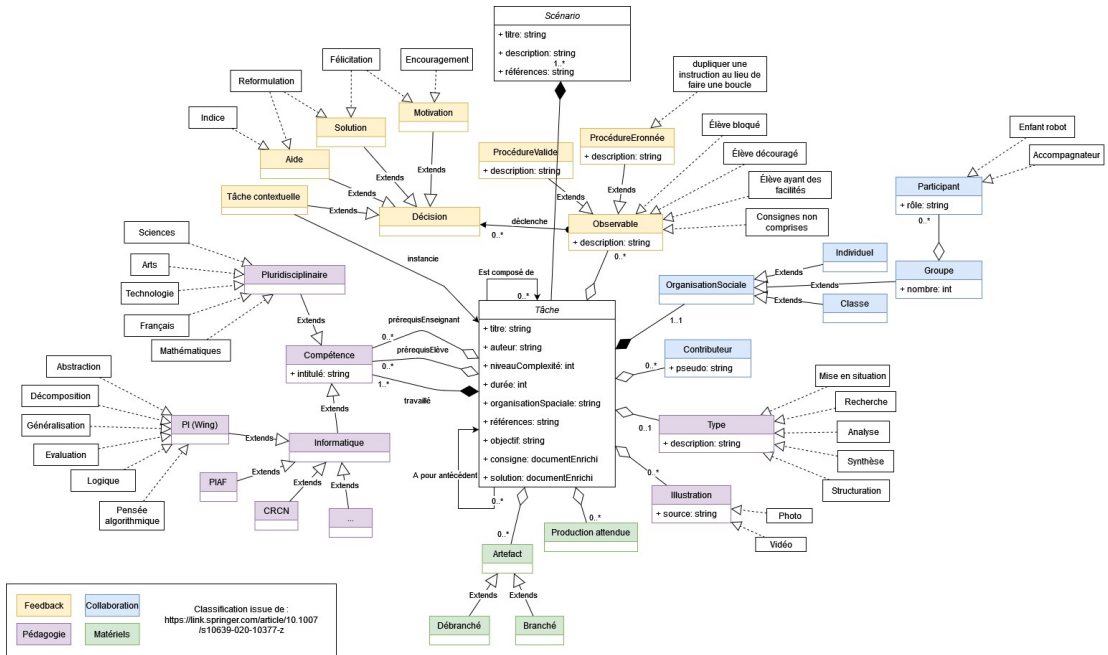


FIGURE 1 – Modèle de description d’un scénario d’enseignement de la PI (Brunet *et al.*, 2020).

Ce modèle (*cf.* figure 1; Brunet *et al.*, 2020), ses différentes composantes (scénario, tâche/activité, compétences, organisation sociale, observable, etc.) et leurs attributs ont été réifiés dans un outil d’aide à la scénarisation : ScenoClasse<sup>8</sup>. ScenoClasse permet aux enseignants de créer, éditer, documenter et agencer des activités d’enseignement de la PI en scénarios puis de publier et d’exporter ces scénarios sous différents formats de sortie (fiche de préparation papier ou sur tablette/téléphone, diaporama, etc.). Ils peuvent également copier et adapter des scénarios créés par d’autres. Chaque scénario rendu public sur la plateforme ScenoClasse peut être enrichi collectivement par des commentaires et des retours d’expériences. Ces fonctionnalités ont été pensées pour permettre à un scénario d’avoir un cycle de vie au-delà du contexte prévu par son auteur.

8. ScenoClasse : <https://scenoclasse.lip6.fr/>, accédé le 12/10/2023

Comme évoqué précédemment, l’objectif de notre recherche est d’assister les enseignants dans le processus de préparation de leurs séances d’introduction à la PI, à travers des rétroactions adaptées à leurs besoins sur la plateforme. Dans cette perspective, il est crucial d’avoir une compréhension des profils des enseignants utilisant la plateforme et de l’évolution de ces profils au cours du temps. Dans le cadre de ce chapitre, nous présentons une première analyse des usages de la plateforme (Kanaan et Yessad, 2023) dont l’objectif est d’identifier des schémas (*patterns*) d’interaction ou des profils d’usages de la plateforme. Il s’agit, par exemple, d’identifier des usages récurrents chez des enseignants expérimentés *versus* des enseignants non expérimentés ou chez des enseignants

qui auraient tendance à modifier des scénarios existants ou à consulter certains types d'informations (compétences travaillées, retour d'expérience des autres enseignants, etc.). Afin d'atteindre cet objectif, nous avons défini un processus d'analyse des traces d'interactions laissées par les enseignants sur la plateforme. Ce processus est organisé en deux étapes. La première étape consiste en un partitionnement des traces d'interactions en plusieurs partitions ou clusters de profils d'usage et la deuxième consiste en l'application d'un algorithme d'extraction de patrons d'usages (ou *pattern mining*) sur chaque cluster issu de la première étape. L'étape de *clustering* a permis d'identifier quatre clusters d'enseignants et l'étape de *pattern mining* a permis de caractériser finement chacun des clusters pour mieux comprendre les usages des enseignants sur la plateforme de scénarisation ScenoClasse.

## Études de la plateforme ScenoClasse

Dans cette recherche, nous avons mené deux études avec deux objectifs différents :

- première étude : évaluer l'utilisabilité de la plateforme ScenoClasse en contexte de formation et en contexte écologique ;
- deuxième étude : analyser les usages des enseignants de primaire participant à l'étude via la fouille de leurs traces d'interactions.

L'objectif de ces deux études est de collecter des données à la fois qualitatives et quantitatives. Pour cela, nous avons mené plusieurs actions dans trois contextes différents :

- Le premier contexte est celui de l'Institut national supérieur du professorat et de l'éducation (INSPE) de l'université du Maine lors de deux ateliers de 90 minutes chacun (mars 2022). Trente-trois participants ont pris part à ces deux ateliers : 25 enseignants de primaire et huit étudiants stagiaires. Ces ateliers ont permis de collecter des données qualitatives sur l'utilisabilité de la plateforme, ses limites et des améliorations possibles. Nous avons également calculé le score d'utilisabilité de ScenoClasse.
- Le deuxième contexte est celui de l'atelier « ScenoClasse : Décrire, visualiser, partager et adapter des scénarios pour l'enseignement de l'informatique à l'école primaire » du colloque Didapro 2022 (mai 2022). Cet atelier nous a permis de réévaluer l'utilisabilité de ScenoClasse après la prise en compte des retours des enseignants du premier contexte. Nous avons également pu tracer les interactions des participants lors de cet atelier et les enregistrer dans une base de données. Les interactions représentent toutes les actions

exécutées par les participants sur ScenoClasse (création d'un nouveau scénario, ajout d'une activité, saisie d'un champ d'information, etc.). Vingt-neuf enseignants ont participé à ces ateliers : 11 enseignants de primaire, 6 formateurs et 12 enseignants stagiaires. Les deux sexes étaient représentés (24 femmes et 5 hommes) et les participants avaient des niveaux d'expérience différents dans l'utilisation des outils numériques dans le cadre professionnel (six participants expérimentés, neuf participants moyennement expérimentés, cinq participants peu expérimentés et neuf participants sans aucune expérience).

- Le troisième contexte a été réalisé en situation écologique avec une classe de CM2 de 22 élèves de l'académie de Versailles. Ce contexte nous a permis d'exploiter la dimension collaborative des scénarios de ScenoClasse pour partager un retour d'expérience. Nous avons ainsi révisé la description du scénario testé et nous avons dégagé pour chacune des tâches/activités du scénario des points de vigilance à partir des difficultés rencontrées à la fois par les élèves et par l'enseignante animatrice de la séance.

## Protocole

Un protocole a été mis au point pour d'une part évaluer l'utilisabilité de la plateforme ScenoClasse et d'autre part permettre la récupération de traces d'interaction pour les analyses envisagées. Nous avons commencé par alimenter la plateforme avec des ressources d'enseignement de la pensée informatique pour des élèves en primaire (cycles 1, 2 et 3). Ensuite, nous avons conçu un questionnaire qualitatif pour recueillir les retours des enseignants sur la plateforme en mettant le focus sur son utilisabilité et les améliorations possibles. Ce questionnaire contenait des questions sur l'expérience du participant, son usage du numérique et son intérêt pour ScenoClasse et incluait les questions du questionnaire standardisé F-SUS (version française du questionnaire *system usability scale*; Gronier et Baudet, 2021). Ce dernier est composé de dix questions avec une échelle de Likert en cinq points de « Pas du tout d'accord » à « Tout à fait d'accord » sur les trois critères d'utilisabilité : l'efficacité, l'efficience et la satisfaction. Nous avons également mis au point un guide<sup>9</sup> pour aider les enseignants à découvrir les fonctionnalités de ScenoClasse. Ce guide était disponible à tout moment sur la plateforme et avait pour objectif de compléter la formation en présentiel qui a été délivrée aux enseignants participant aux études.

Pendant les différents ateliers réalisés lors du colloque Didapro 2022 (le deuxième contexte), nous avons à chaque fois commencé par présenter la plateforme aux participants. Ensuite, nous leur

9. Questionnaire pour ScenoClasse : <https://scenoclasse.lip6.fr/blog/guides/>, accédé le 12/10/2023

avons demandé d'explorer librement la plateforme puis de travailler sur la plateforme pour créer un nouveau scénario ou bien cloner et modifier un scénario existant. Enfin, les participants ont été sollicités pour remplir le questionnaire.

### **Retours qualitatifs et test d'utilisabilité (contextes un et deux)**

À l'issue du questionnaire soumis aux participants dans les deux premiers contextes, le test d'utilisabilité de ScenoClasse nous a remonté des scores positifs (le score F-SUS du premier contexte = 61,40/100 et le score F-SUS du deuxième contexte = 70,83/100). Les participants ont clairement exprimé leur intention de continuer à utiliser ScenoClasse et de recommander la plateforme à leurs collègues enseignants. Cependant, ils ont pointé certains problèmes d'utilisabilité et ont suggéré des améliorations. En particulier, les participants du premier contexte ont demandé la mise en place d'un système de gestion de comptes personnels, l'adaptation du vocabulaire utilisé à celui des enseignants, et avoir la possibilité de relier certains scénarios entre eux. Une nouvelle phase de développement de ScenoClasse a permis d'intégrer ces besoins et d'améliorer ainsi le score F-SUS du test d'utilisabilité du deuxième contexte. Vingt-trois des 29 expérimentateurs qui ont participé au deuxième contexte ont évalué ScenoClasse comme une plateforme conviviale et facile à prendre en main.

### **Retour du processus d'analyse de traces à deux étapes (contexte deux)**

Lors de la deuxième étude réalisée dans le deuxième contexte, nous avons collecté et analysé les traces de 51 sessions de travail différentes.

### **Modèle de traces**

Les actions des participants ont été modélisées sous forme de traces de différents types et niveaux de granularité et enregistrées dans une base de données. Nous distinguons des actions sur les scénarios (créer un scénario, cloner un scénario, supprimer un scénario, imprimer un scénario, etc.), des actions sur les activités composant un scénario (créer une activité, supprimer une activité, réorganiser les liens entre des activités au sein d'un même scénario, etc.), des actions sur les champs d'une activité (mettre à jour les compétences travaillées dans une activité, la description d'une activité, les ressources nécessaires pour une activité, etc.) et enfin

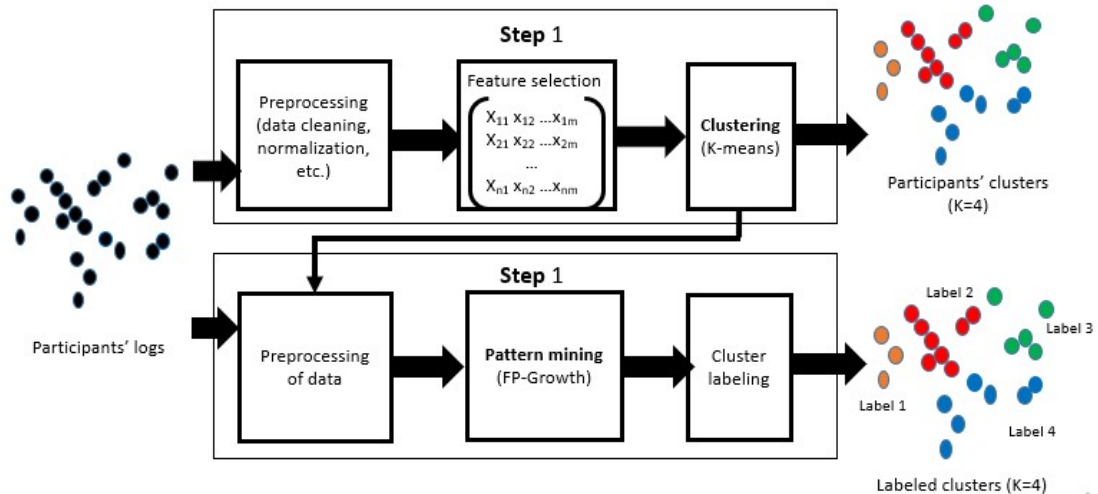


FIGURE 2 – Processus d'analyse à deux étapes.

des actions au niveau de la plateforme (créer un compte, chercher un scénario, demander la publication d'un scénario, etc.).

### Processus d'analyse à deux étapes pour identifier les comportements d'usage de la plateforme par les enseignants de primaire

Nous avons défini et appliqué un processus à deux étapes pour identifier les usages des enseignants sur la plateforme ScenoClasse à partir de leurs traces d'interaction (cf. figure 2). D'abord, nous avons utilisé l'algorithme de *clustering K-means* (MacQueen, 1967) pour créer des groupes de comportements d'usage suffisamment différents.

Pour cela, nous avons prétraité les traces en normalisant les valeurs des attributs et avons sélectionné les attributs caractérisant chaque session de travail sur ScenoClasse. Cette sélection d'attributs (*feature selection*) a été une étape critique que nous avons réalisée itérativement en testant différents attributs. Parmi les attributs testés, trois attributs (nombre de scénarios clonés, nombre de solutions consultées, nombre d'activités explorées au-delà d'un certain temps) ont permis des regroupements maximisant la distance entre les centroïdes des clusters. Ainsi, les sessions de travail caractérisées par ces trois attributs ont constitué l'entrée de l'algorithme de *clustering*. Ensuite, nous avons utilisé l'algorithme d'extraction de motifs séquentiels *generalized sequential pattern* (Srikant et Agrawal, 1996) pour caractériser plus précisément chaque cluster identifié.



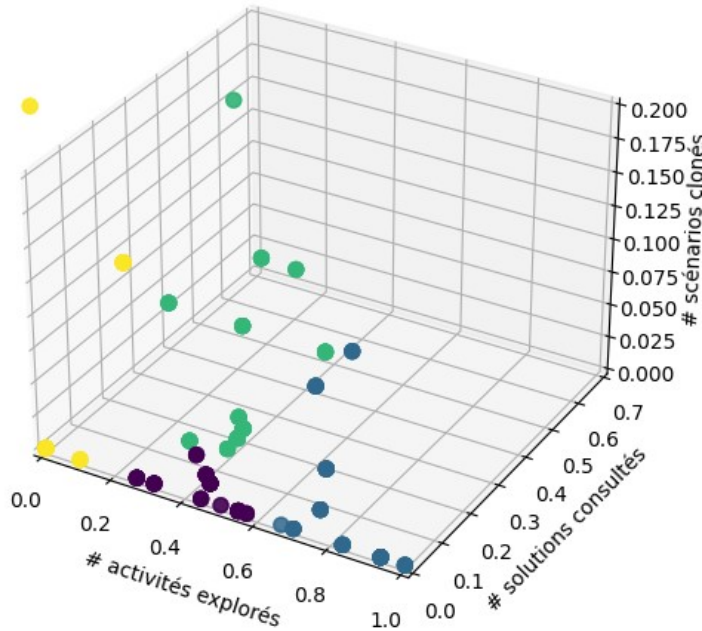


FIGURE 3 – Le partitionnement des différentes traces en clusters. Chaque cluster est coloré différemment et est caractérisé par les attributs utilisés, à savoir, le nombre d'activités explorées, le nombre de scénarios clonés et le nombre de solutions consultées.

### Résultat de l'analyse

Certains participants ont produit deux sessions de travail différentes. Nous avons gardé toutes les sessions de travail même lorsque celles-ci appartenaient au même participant car notre objectif était de détecter des *patterns* d'usages différents et non d'identifier la proportion des participants de tel ou tel usage. À l'issue de la première étape d'analyse, nous avons obtenu quatre clusters de sessions de travail maximisant la distance entre les centroïdes des clusters (*average centroid distance* = -0,763) :

- cluster zéro (cluster jaune) : regroupant des participants qui sont davantage intéressés par le clonage des scénarios en vue de les modifier et moins par la consultation en lecture de scénarios existants ;
- cluster un (cluster bleu) : regroupant des participants découvrant la plateforme en testant ses différentes fonctionnalités sans prendre le temps d'explorer une fonctionnalité en particulier ;
- cluster deux (cluster violet) : regroupant des participants qui ont passé du temps à découvrir et explorer les différentes activités et leurs descriptions et parfois leurs solutions ;

- cluster trois (cluster vert) : regroupant des participants qui ont passé du temps à regarder les solutions des activités sans forcément consulter les autres informations décrivant les activités.

L'application de l'algorithme *generalized sequential pattern mining* (GPS) sur les traces de chaque cluster séparément a confirmé l'interprétation que nous avons eu de chacun des clusters. Le contexte de la deuxième étude, à savoir, des ateliers courts et assez guidés n'a pas permis une variation importante dans les données. Les trois attributs sélectionnés dans l'étape *clustering* permettent à eux seuls d'expliquer les différents comportements d'usage des 29 participants de l'étude. Cette étape de *pattern mining* du processus à deux étapes que nous avons défini, indépendamment du contexte de l'étude, serait davantage utile dans un contexte où les traces seraient plus nombreuses avec des attributs plus nombreux pour les caractériser et donc potentiellement des clusters plus difficiles à interpréter.

Néanmoins, le *pattern mining* a permis, malgré le contexte assez contraint de l'étude, de remonter des informations complémentaires au *clustering* sur la nature des actions réalisées et leurs fréquences pour les participants de chacun des clusters d'usages. Nous synthétisons la répartition des actions des participants dans chacun des clusters comme suit :

- les participants du cluster zéro qu'on pourrait qualifier de « producteurs / adaptateurs » ont essentiellement cloné des scénarios et commencé à les modifier et ne se sont pas intéressés à les exporter pour les visualiser ou consulter les commentaires laissés par d'autres utilisateurs. Ces participants ont peut-être une expérience de ce type de scénarios de la pensée informatique ;
- les participants du cluster un qu'on pourrait qualifier de « prospecteurs » ont exploré plusieurs fonctionnalités de la plateforme avec une répartition quasi identique entre les différentes actions de leur session de travail. Ce cluster est probablement le plus prévisible des quatre puisqu'il est cohérent avec le contexte de l'étude. Les participants ne sont pas familiers avec la plateforme donc il apparaît normal qu'ils découvrent d'abord celle-ci avant de commencer à l'utiliser pour répondre à des besoins pédagogiques ;
- les participants du cluster deux qu'on pourrait qualifier de « consommateurs » ont beaucoup consulté les activités disponibles en s'attardant sur leurs descriptions sans les modifier ;
- enfin, les participants du cluster trois peuvent être des « débutants » qui ont principalement consulté les solutions

disponibles des activités sans aller chercher d'autres informations sur celles-ci (compétences travaillées, prérequis, etc.).

### **Retour d'utilisabilité d'un scénario en contexte écologique (contexte trois)**

10. SPY : <https://spy.lip6.fr>,  
accédé le 12/10/2023

Dans cette expérimentation nous avons testé un scénario de la plateforme exploitant le jeu sérieux SPY<sup>10</sup> (Muratet, 2023). SPY est un jeu sérieux d'apprentissage sur le thème de la pensée informatique. Il a été conçu pour un public d'élèves de cycle 3 (CM1, CM2, 6<sup>e</sup>). Le principe du jeu est de programmer un ou plusieurs agents à l'aide d'actions simples afin de les déplacer sur une grille vers des positions particulières. Ces actions sont représentées sous forme de blocs que l'élève doit agencer en séquences.

11. Scénario SPY décrit dans Sceno-  
Classe : [https://scenoclasse.lip6.fr/  
editeur/scenario/643575d71d35e](https://scenoclasse.lip6.fr/editeur/scenario/643575d71d35e),  
accédé le 12/10/2023

Nous distinguons bien ici les scénarios ludiques définis dans le jeu SPY du scénario pédagogique décrit sur la plateforme ScenoClasse. Pour autant ces deux types de scénarios sont complémentaires. En effet, décrire une situation pédagogique avec ScenoClasse exploitant la ressource SPY permet d'enrichir le scénario ludique par des compléments, des vigilances, des recommandations de durée... Nous avons ainsi décrit, dans l'application ScenoClasse, un scénario pédagogique pour SPY incluant trois des scénarios ludiques du jeu<sup>11</sup>. Afin de fournir les éléments attendus par les comportements qualifiés de « consommateurs » du cluster deux, nous avons intégré dans le scénario pédagogique les compétences PIAF travaillées dans le jeu (Muratet, 2023) ainsi que la solution des niveaux.

Lors de cette expérimentation, trois scénarios ludiques du jeu ont été proposés aux élèves par l'enseignante. Le premier intitulé « Explorateur » est composé de neuf niveaux où l'élève ne manipule que des blocs d'action (« Avancer », « Pivoter à gauche », « Pivoter à droite », « Attendre »...). Le deuxième scénario, intitulé « Collaborateur », est composé de 12 niveaux. Dans ce scénario, l'élève doit faire collaborer deux robots. Il doit notamment apprendre à nommer correctement les zones de programmation afin de contrôler les bons robots. Le troisième scénario, intitulé « Répétiteur », est composé de dix niveaux. Ce scénario est centré sur la notion de répétition. Ici les blocs actions sont limités pour forcer le joueur à utiliser le bloc « Répéter plusieurs fois ».

SPY est donc un jeu sérieux sur le thème de la pensée informatique qui propose des scénarios ludiques clé en main qui peuvent être directement utilisables par des enseignants dont les comportements ont été classés dans les clusters « débutants » ou « consommateurs ». L'éditeur de scénario ludique et l'éditeur

de niveau intégrés au jeu sont aussi des ressources qui peuvent répondre aux enseignants identifiés comme ayant un comportement « producteurs/adaptateurs ». Ces éditeurs permettent aux enseignants de modifier les scénarios ludiques existants et créer leurs propres scénarios/niveaux.

Ce scénario pédagogique, composé des trois scénarios ludiques de SPY, a donc été expérimenté dans une classe de CM2 composée de 22 élèves. Cinq séances d'une heure ont été organisées. Pour chaque séance, le groupe classe était divisé en deux groupes pendant les 30 premières minutes, 11 élèves jouaient au jeu SPY sur ordinateur (un élève par poste) et les 11 autres faisaient des activités de mathématiques ou de français sur tablette. Au bout de 30 minutes, les deux groupes intervertissaient. À l'issue de cette expérimentation en contexte écologique, nous avons exploité la dimension collaborative des scénarios de ScenoClasse pour partager le retour d'expérience des utilisateurs. Nous avons ainsi révisé la description du scénario pédagogique et nous avons dégagé pour chacune des tâches du scénario des points de vigilance à partir des difficultés rencontrées à la fois par les élèves et l'enseignante animatrice de la séance. Par exemple, nous avons explicité des informations portant sur les savoirs comme la différence entre actions atomiques et action non-atomiques sur le niveau sept du scénario « Explorateur », sur les aspects ludiques comme le fait que deux robots ne peuvent pas utiliser la même case sur le niveau huit du scénario « Collaborateur », ou portant sur les manipulations de l'artefact afin d'expliciter certaines procédures comme la création de nouvelles zones de programmation sur le niveau 11 du scénario « Collaborateur ». À l'issue de cette expérimentation nous avons donc identifié 12 points de vigilance qui ont permis d'enrichir la description du scénario pédagogique de SPY sur ScenoClasse pour être partagé aux autres utilisateurs du scénario.

## Conclusion

Dans ce chapitre, nous présentons une contribution de recherche qui a pour objectif de soutenir les enseignants de primaire débutant dans l'enseignement de la pensée informatique. D'une part, nous avons présenté un modèle de scénarisation qui a été réifié dans une plateforme de scénarisation ScenoClasse. Deux études ont été menées dans trois contextes différents : d'une part pour évaluer l'utilisabilité de ScenoClasse et d'autre part pour identifier des comportements d'usage de la plateforme ScenoClasse par des enseignants de primaire. Les retours des enseignants lors du premier contexte ont montré un bon score SUS (score F-SUS = 61,40/100) et ont permis, après une nouvelle phase de développement, d'améliorer l'utilisabilité de ScenoClasse dans le contexte

deux (score F-SUS = 70,83/100). Le deuxième contexte expérimental a permis de formaliser un processus d'analyse à deux étapes (*clustering* puis extraction de motifs séquentiels) qui a servi à analyser les traces laissées par les enseignants et à identifier quatre profils d'usages de la plateforme : « producteurs/adaptateurs », « prospecteurs », « consommateurs » et « débutants ». Le troisième contexte expérimental a permis de montrer l'utilisabilité d'un scénario ScenoClasse en situation écologique et a permis de dégager des points de vigilance pour enrichir le scénario mis à disposition de la communauté enseignante via la plateforme.

Le travail d'analyse des traces amorcé dans le cadre du projet IE CARE de l'ANR devra être poursuivi sur les données récupérées d'autres contextes, en particulier, des contextes où des enseignants utilisent ScenoClasse pour préparer des séances d'enseignement de la PI effectivement mises en œuvre avec leurs élèves. Aussi, la dimension temporelle des traces devrait être considérée pour étudier l'évolution des usages des enseignants. Un enseignant changerait-il de cluster suite à une pratique régulière de la plateforme et dans quelles conditions ? Quelles seraient les trajectoires de profils de ces enseignants sur la plateforme ?

Au-delà des profils identifiés dans le contexte de l'expérimentation contrôlée menée dans le cadre du projet IE CARE, notre contribution porte davantage sur le processus d'analyse que nous avons explicité que sur les résultats obtenus. Ce processus d'analyse pourrait servir de cadre pour d'autres expérimentations à venir dont l'objectif serait d'identifier des *patterns* dans d'autres contextes d'usages (préparation de séances, formation professionnelle ou formation initiale des professeurs, etc.). Ces derniers pourraient servir à apporter une aide adaptée aux enseignants de l'école primaire. Il pourrait être envisagé d'enrichir les résultats de ce processus avec des données qualitatives afin de mieux interpréter les résultats voire orienter le processus lui-même en guidant par exemple la sélection d'attributs (*features*) en entrée du processus.

## Références

- Abiteboul, S., Archambault, J.-P., Balagué, C., Baron, G.-L., Berry, G., Doweck, G., de la Higuera, C., Nivat, M., Tort, F., et Viéville, T. (2015). *L'enseignement de l'informatique en France — Il est urgent de ne plus attendre*, Académie des sciences, p. 35. <https://www.academie-sciences.fr/fr/Rapports-ouvrages-avis-et-recommandations-de-l-Academie/l-enseignement-de-l-informatique-en-france-il-est-urgent-de-ne-plus-attendre.html>
- Alayrangues, S., Peltier, S., et Signac, L. (2017). Informatique débranchée : construire sa pensée informatique sans ordinateur, *colloque Mathématiques en cycle 3 IREM de Poitiers*, p. 216-226, Poitiers, France. <https://hal.science/hal-01868132>
- Anseel, F., Lievens, F., et Schollaert, E. (2009). Reflection as a strategy to enhance task performance after feedback, *Organizational Behavior and Human Decision Processes*, vol. 110, n° 1, p. 23-35. <https://doi.org/10.1016/j.obhdp.2009.05.003>
- Bardach, L., Klassen, R. M., Durksen, T. L., Rushby, J. V., Bostwick, K. C. P., et Sheridan, L. (2021). The power of feedback and reflection : Testing an online scenario-based learning intervention for student teachers, *Computers & Education*, vol. 169, p. 104-194. <https://doi.org/10.1016/j.compedu.2021.104194>
- Bennacer, I. (2022). Teaching analytics : Support for the evaluation and assistance in the design of teaching through artificial intelligence, thèse de l'université du Mans. <https://theses.hal.science/tel-03935709>
- Brennan, K., et Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA 2012)*, vol. 1, n° 25. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Brinkerhoff, J. (2006). Effects of a long-duration, professional development academy on technology skills, computer self-efficacy, and technology integration beliefs and practices, *Journal of Research on Technology in Education*, vol. 39, n° 1, p. 22-43. <https://doi.org/10.1080/15391523.2006.10782471>
- Brown, A. L., Lee, J., et Collins, D. (2015). Does student teaching matter? Investigating pre-service teachers' sense of efficacy and preparedness, *Teaching Education*, vol. 26, n° 1, p. 77-93. <https://doi.org/10.1080/10476210.2014.95766>
- Brunet, O., Yessad, A., Muratet, M., et Carron, T. (2020). Vers un modèle de scénarisation pour l'enseignement de la pensée informatique à l'école primaire, *Didapro 8 – DidaSTIC*. <https://hal.science/hal-02496191>

- Chen, G. (2020). A visual learning analytics (VLA) approach to video-based teacher professional development : Impact on teachers' beliefs, self-efficacy, and classroom talk practice, *Computers & Education*, vol. 144, p. 103-670. <https://doi.org/10.1016/j.compedu.2019.103670>
- De Coninck, K., Valcke, M., Ophalvens, I., et Vanderlinde, R. (2019). « Bridging the theory-practice gap in teacher education : The design and construction of simulation-based learning environments », dans K. Hellmann, J. Kreutz, M. Schwichow, et K. Zaki (dir.), *Kohärenz in der Lehrerbildung*, Springer Fachmedien Wiesbaden, p. 263-280. [https://doi.org/10.1007/978-3-658-23940-4\\_17](https://doi.org/10.1007/978-3-658-23940-4_17)
- Denning, P. J. (2017). Remaining trouble spots with computational thinking, *Communications of the ACM*, vol. 60, n° 6, p. 33-39. <https://doi.org/10.1145/2998438>
- Dillenbourg, P. (2015). *Orchestration graphs : Modeling scalable education*, EPFL press.
- Goggins, S. (2012). Group informatics : A multi-domain perspective on the development of teaching analytics, *CEUR Workshop Proceedings*, vol. 894.
- Gronier, G., et Baudet, A. (2021). Psychometric evaluation of the F-SUS : Creation and validation of the French version of the system usability scale, *International Journal of Human-Computer Interaction*, vol. 37, n° 16, p. 1571-1582. <https://doi.org/10.1080/10447318.2021.1898828>
- Holden, H., et Rada, R. (2011). Understanding the influence of perceived usability and technology self-efficacy on teachers' technology acceptance, *Journal of Research on Technology in Education*, vol. 43, n° 4, p. 343-367. <https://doi.org/10.1080/15391523.2011.10782576>
- Hursen, C., et Fasli, F. G. (2017). Investigating the efficiency of scenario based learning and reflective learning approaches in teacher education, *European Journal of Contemporary Education*, vol. 6, n° 2, p. 264-279. <https://doi.org/10.13187/ejced.2017.2.264>
- Kanaan, M., et Yessad, A. (2023). « A Two-Step Process for Analysing Teacher's Behaviors Using a Scenario-Based Platform », dans C. S. González-González, B. Fernández-Manjón, F. Li, F. J. García-Peñalvo, F. Sciarone, M. Spaniol, A. García-Holgado, M. Area-Moreira, M. Hemmje, et T. Hao (dir.), *Learning Technologies and Systems*, Springer International Publishing, vol. 13 869, p. 53-62. [https://doi.org/10.1007/978-3-031-33023-0\\_5](https://doi.org/10.1007/978-3-031-33023-0_5)
- Klassen, R. M., Rushby, J. V., Maxwell, L., Durksen, T. L., Sheridan, L., et Bardach, L. (2021). The development and testing of an online scenario-based learning activity to prepare preservice teachers

for teaching placements, *Teaching and Teacher Education*, vol. 104, p. 103-385. <https://doi.org/10.1016/j.tate.2021.103385>

Komis, V., et Misirli, A. (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle : une étude de cas basée sur le jouet programmable Bee-Bot, *Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques*, vol. 271. <https://edutice.hal.science/edutice-00676143>

Komis, V., Tzavara, A., Karsenti, T., Collin, S., et Simard, S. (2013). Educational scenarios with ICT : An operational design and implementation framework, *Society for Information Technology & Teacher Education International Conference*, p. 3244-3251. [https://www.researchgate.net/publication/289988466\\_Educational\\_scenarios\\_with\\_ICT\\_an\\_operational\\_design\\_and\\_implementation\\_framework](https://www.researchgate.net/publication/289988466_Educational_scenarios_with_ICT_an_operational_design_and_implementation_framework)

Kradolfer, S., Dubois, S., Riedo, F., Mondada, F., et Fassa, F. (2014). « A Sociological contribution to understanding the use of robots in schools : The thymio robot », dans M. Beetz, B. Johnston, et M.-A. Williams (dir.), *Social Robotics*, Springer International Publishing, p. 217-228. [https://doi.org/10.1007/978-3-319-11973-1\\_22](https://doi.org/10.1007/978-3-319-11973-1_22)

Ku, O., Liang, J.-K., Chang, S.-B., et Wu, M. (2018). Sokrates Teaching Analytics System (STAS) : An Automatic Teaching Behavior Analysis System for Facilitating Teacher Professional Development, *26th International Conference on Computers in Education*, Metro Manila, Philippines.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., et Werner, L. (2011). Computational thinking for youth in practice, *ACM Inroads*, vol. 2, n° 1, p. 32-37. <https://doi.org/10.1145/1929887.1929902>

MacQueen, J. (1967). « Some methods for classification and analysis of multivariate observations », dans *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, vol. 1, n° 5.1, p. 281-298.

Miljanovic, M. A., et Bradbury, J. S. (2018). « A Review of Serious Games for Programming », dans S. Göbel, A. Garcia-Agundez, T. Tregel, M. Ma, J. Baalsrud Hauge, M. Oliveira, T. Marsh, et P. Caserman (dir.), *Serious Games*, Springer International Publishing, vol. 11243, p. 204-216. [https://doi.org/10.1007/978-3-030-02762-9\\_21](https://doi.org/10.1007/978-3-030-02762-9_21)

Muratet, M. (2023). Comment caractériser et analyse les compétences de la pensée informatique d'un jeu sérieux ?, *EIAH 2023*, p. 178-180. <https://hal.science/hal-04246906/document>

Ndukwe, I. G., et Daniel, B. K. (2020). Teaching analytics, value and tools for teacher data literacy : A systematic and tripartite



approach, *International Journal of Educational Technology in Higher Education*, vol. 17, n° 1, p. 22. <https://doi.org/10.1186/s41239-020-00201-6>

Papert, S. (1993). *Mindstorms : Children, computers, and powerful ideas* (2nd ed.), Basic Books.

Parmentier, Y. (2018). Enseigner la pensée informatique à l'école primaire : formation initiale et continue des professeurs, *Atelier « Organisation et suivi des activités d'apprentissage de l'informatique : outils, modèles et expériences » RJC-EIAH*. <https://hal.science/hal-01762626>

Pólya, G., et Conway, J. H. (1957). *How to solve it : A new aspect of mathematical method*, Princeton University Press.

Romero, M., Lille, B., Viéville, T., Duflot-Kremer, M., Smet, C. D., et Belhassein, D. (2018). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché, *Educode - Conférence internationale sur l'enseignement au numérique et par le numérique*. <https://inria.hal.science/hal-01861732>

Saddoug, H., Rahimian, A., Marne, B., Muratet, M., Sehaba, K., et Jolivet, S. (2022). Review of the Adaptability of a Set of Learning Games Meant for Teaching Computational Thinking or Programming in France, *Proceedings of the 14th International Conference on Computer Supported Education*, p. 562-569. <https://doi.org/10.5220/0011126400003182>

Schrader, P., Mariappan, J., et Shih, A. (2004). Scenario Based Learning Approach In Teaching Statics, *2004 Annual Conference Proceedings*, 9.1083.1-9.1083.7. <https://doi.org/10.18260/1-2-13347>

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

Seren Smith, M., Warnes, S., et Vanhoestenbergh, A. (2018). « Scenario-based learning », dans J. P. Davies et N. Pachler (dir.), *Teaching and Learning in Higher Education : Perspectives from UCL*, UCL IOE Press, p. 144-156. <https://www.ucl-ioe-press.com/ioe-content/uploads/2018/05/Teaching-and-Learning-in-Higher-Education.pdf>

Sergis, S., et Sampson, D. G. (2017). « Teaching and Learning Analytics to Support Teacher Inquiry : A Systematic Literature Review », dans A. Peña-Ayala (dir.), *Learning Analytics : Fundamentals, Applications, and Trends*, Springer International Publishing, vol. 94, p. 25-63. [https://doi.org/10.1007/978-3-319-52977-6\\_2](https://doi.org/10.1007/978-3-319-52977-6_2)

Spach, M. (2019). Activités robotiques à l'école : approches de pratiques d'enseignement et effets sur les apprentissages, *Recherches en didactiques*, vol. 28, n° 2, p. 68-87. <https://doi.org/10.3917/rdid.028.0068>

- Srikant, R., et Agrawal, R. (1996). « Mining sequential patterns : Generalizations and performance improvements », dans P. Apers, M. Bouzeghoub, et G. Gardarin (dir.), *Advances in Database Technology—EDBT '96*, Springer Berlin Heidelberg, vol. 1057, p. 1-17. <https://doi.org/10.1007/BFb0014140>
- Storey, M.-A., Phillips, B., Maczewski, M., et Wang, M. (2002). Evaluating the usability of web-based learning tools, *Journal of Educational Technology & Society*, vol. 5, n° 3, p. 91-100.
- Vahldick, A., Mendes, A. J., et Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies, *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, p. 1-7. <https://doi.org/10.1109/FIE.2014.7044114>
- Watson, G. (2006). Technology Professional Development : Long-Term Effects on Teacher Self-Efficacy, *Journal of Technology and Teacher Education*, vol. 14, n° 1, p. 151-166.
- Wing, J. M. (2006). Computational thinking, *Communications of the ACM*, vol. 49, n° 3, p. 33-35. <https://doi.org/10.1145/1118178.1118215>

Pour citer ce chapitre :

Yessad, Amel, Muratet, Mathieu, et Carron, Thibault (2024). « Assister les enseignants du primaire dans l'enseignement de la pensée informatique. Une approche basée sur la scénarisation et vers les *teaching analytics* », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Refay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 211-231. <https://doi.org/10.53480/2024iecare11y>



# POSTFACE



# En guise de postface : de modèles d'analyse à un modèle de conception de formations à l'informatique

Eric BRUILLARD<sup>1</sup>

1. Université Paris Cité, Laboratoire  
EDA, 75006 Paris, France

L'informatique, côté éducation, est un sujet difficile à circonscrire et à scolariser. Deux options majeures sont présentes, correspondant à deux visions éloignées de l'informatique et de son enseignement dans le cadre scolaire :

- Si on la considère en tant que discipline scolaire, se référer aux disciplines existantes invite le plus souvent à la regarder comme une transposition de certains aspects de la discipline universitaire, avec quelques contraintes ou opportunités imposées ou offertes dans les prescriptions autour de projets. Le cœur en est la programmation, matinée d'un discours autour de la pensée informatique permettant à l'informatique de revendiquer plus qu'elle-même dans les bénéfices liés à son enseignement (Drot-Delange *et al.*, 2019). S'ajoutent également des aspects plutôt sociétaux. Mais comment développer des éléments de formation autour d'informatique et société, qui étaient présents dans l'option informatique dans les années 1980 (Baron, 1989), mais qui sont éloignées de la discipline universitaire<sup>1</sup> ?
- Si on y voit un champ mêlant sciences, technologies et utilisations (Fluckiger, 2019), les références universitaires deviennent multiples, articulant de manière plutôt chaotique, des visions scientifiques, technologiques, sociales et humaines. Par exemple, qu'est-ce que comprendre un algorithme : être capable de le transcrire dans un langage de programmation, connaître sa finalité orientant son fonctionnement, comprendre les choix qu'il effectue, en donner une description fine, envisager les conséquences de son utilisation dans différents contextes. . .

1. Une raison des difficultés de la scolarisation d'un enseignement informatique et société est le manque d'exercitation directement associée comme peut l'être la programmation.

Obscurcissant le questionnement, des pressions fortes depuis plus d'une dizaine d'années sur la nécessité de l'enseignement du « code » (Boissière et Bruillard, 2021, p. 90) ont relancé l'intérêt d'enseigner l'informatique, en se centrant sur la programmation et

le codage ; le confinement imposé suite à l'épidémie de Covid et les discours et injonctions sur la « continuité pédagogique », ont mis l'accent sur l'enseignement à distance, conduisant à renforcer des confusions réduisant la question numérique à celle de la distance ; l'explosion des intelligences artificielles génératives, type ChatGPT, multiplie les discours, sans clarifier les liens possibles avec des enseignements de l'informatique.

Le projet IE CARE de l'Agence nationale de la recherche (ANR) s'est inscrit dans la dynamique précédant l'épidémie de Covid dans une vision ouverte de l'informatique à l'école, mais avec en toile de fond une vision dominante de l'enseignement de l'informatique correspondant, voire se limitant, à l'enseignement du code ou de l'algorithmique. Cela n'a pas été sans tension et le projet, débuté dans un contexte favorable à l'enseignement de l'informatique à l'école et au collège, se termine dans un contexte qui l'est beaucoup moins, comme le décrit Georges-Louis Baron dans sa préface.

Nous allons d'abord brièvement reprendre quelques aspects du projet IE CARE, dont le présent ouvrage, présente une partie des réalisations. Nous interrogerons les questionnements choisis, articulés autour de modèles d'analyse de la situation pour des chercheurs, et nous proposerons un modèle de conception d'enseignements informatiques que l'on peut également utiliser comme modèle d'analyse de formations existantes.

## **IE CARE, un projet pour accompagner l'apprentissage de l'informatique**

L'acronyme IE CARE se développe de la manière suivante : « Informatique à l'école : conceptualisations, accompagnement, ressources ». Acceptant la prise en compte d'une triple nature de l'informatique – (1) une science, (2) une (des) technologie(s), (3) des utilisations ou des usages, la recherche IE CARE s'est basée sur trois axes de développement, reliés par une tâche transversale de mise en cohérence :

1. délimiter un ensemble de contenus informatiques enseignables à l'école primaire ;
2. concevoir et modifier des scénarios pédagogiques et des ressources (des valises numériques et technologiques) pour soutenir les pratiques d'enseignement et d'apprentissage ;
3. étudier les cadres d'accompagnement mis en place par et pour les enseignants et les formateurs en informatique.

Le point 1 se focalise sur les contenus, le point 2 sur les situations en associant les ressources et le point 3 concerne le système enseignant. Cette articulation en trois tâches donne une idée de l'intérêt des chercheurs, dans l'élaboration de contenus d'enseignement, dans la conception de dispositifs, dans la compréhension des cadres de mises en œuvre.

Il s'agit bien d'une vision large de l'informatique qui n'est pas construite à partir d'une transformation de l'architecture de l'informatique en tant que discipline universitaire. On articule, sans les opposer, concepts et pratiques, sciences et technologies, conception et usages, consommation et production (*prosumer*). Une question sous-jacente est « qu'est-ce que l'informatique scolaire ? » ou « qu'est-ce que faire de l'informatique ? ». Des réponses différentes peuvent être apportées par des experts, des enseignants, des élèves, des parents. . .

Le logo du projet IE CARE (figure 1) montre une plume et une aile déployée avec un disque orange en arrière-plan. Une référence à Icare, avec deux sortes de plumes, l'informatique comme technologie d'écriture et celle qui permet d'atteindre le soleil. Quand on l'attache à un fond, c'est le bleu, couleur caractéristique du numérique (Bruillard, 2019). Un lien entre la plume pour écrire, la plume de l'aile pour voler dans le cadre contemporain, souligné par la couleur bleue et les éclairs blancs. Le fond plus froid permet à Icare de continuer à voler sans encombre, ou peut-être va-t-il se noyer dans l'« océan numérique » ?

Soulignons d'ailleurs que les partenaires grecs du projet, Vassilis Komis et Sevastiani Touloupaki, sont d'origine crétoise. Or, la légende rapporte qu'Icare, emprisonné avec son père Dédale, a voulu fuir la Crète :

Dédale eut l'idée, pour fuir la Crète, de fabriquer des ailes semblables à celles des oiseaux, confectionnées avec de la cire et des plumes. Il met en garde son fils, lui interdisant de s'approcher trop près de la mer, à cause de l'humidité, et du Soleil, à cause de la chaleur. Mais Icare appréciait la vue et voulut en voir plus. Grisé par le vol, il oublia l'interdit et prit de plus en plus d'altitude. La chaleur fait fondre la cire jusqu'à ce que ses ailes finissent par le trahir. Il meurt précipité dans la mer qui porte désormais son nom : la mer Icarienne.

Prémonition d'une vision émancipatrice de l'informatique qui a sombré dans une scolarisation balbutiante ?



FIGURE 1 – Logo et affiche du projet IE CARE.



## Trois : un nombre caractéristique d'une approche ?

On vient de commenter des images du projet, qu'en est-il de sa musique : une valse à trois temps ?

En reprenant l'acronyme du projet IE CARE (Conceptualisations, Accompagnement, REssources), la vision de l'informatique (science, technologie, utilisations) et les axes, apparaît le nombre trois, que l'on retrouve également dans l'ouvrage dont le titre reprend la forme de l'acronyme : « enseigner et apprendre l'informatique à l'école : pratiques, représentations et ressources », avec aussi un triptyque mais dans lequel seul le mot « ressources » est commun. Il y a trois parties (enseigner, apprendre, former).

Le choix de mots et leur évolution révèlent des éléments du positionnement initial du projet et de son opérationnalisation. Même si l'acronyme initial désigne plutôt les objets investigués par la recherche et dans le livre plutôt les objets au plus près des méthodes utilisées, les permanences et disparitions sont intéressantes et significatives.

La disparition des mots « conceptualisation » et « accompagnement » indique que les membres du projet n'ont pas pu se centrer sur ces deux processus (interne et externe), d'une part en raison du confinement qui a limité les suivis, d'autre part en raison de leurs intérêts de recherche, mais également pour des raisons moins circonstanciées et plus profondes. En termes d'appariement, « représentation » se substitue à « conceptualisation » et « pratique » (ou « activité ») à « accompagnement »<sup>2</sup>.

La disparition de « conceptualisation » peut être interprétée par la place encore très instable et mal définie de l'informatique dans le scolaire : les notions censées être acquises ne sont toujours pas claires et face à un flou persistant, étudier les représentations des différents acteurs (formateurs, enseignants, élèves) devient une approche importante. Il y a cependant des processus d'utilisation de robots, de programmation qui ont été observés, par exemple, dans les travaux de Grugier et de Nogry (en partie dans cet ouvrage), également dans la thèse de Touloupaki (2023). Dans ces travaux, on a collecté ce que les personnes font ou disent faire (observations et entretiens) et ont fait (au travers des traces).

La disparition de l'« accompagnement » témoigne en partie de l'absence de la formation continue et d'une évolution générale vers la « plateformes » : à un processus humain répété et local auprès des acteurs se substitue une offre accessible sur une plate-forme fournissant aide et conseils, scénarios voire activités que peuvent mener les acteurs (élèves et enseignants). C'est cohérent avec la persistance du mot « ressources », dans une

2. Le décès de François Villemonteix explique aussi la disparition du mot « accompagnement ». Il était très attaché à cette idée, mais les personnes du projet y étaient sans doute moins sensibles (cf. préface).

large palette d'acceptations, notamment pour les études menées par les chercheurs : analyser les ressources proposées, les choix de ressources par les enseignants et les formateurs ; développer des ressources et des plateformes de ressources, etc.

Le présent ouvrage présente une grande diversité de recherches contribuant à documenter les différents aspects du paysage, des élèves au ministère.

- Élèves : leurs discours, leurs représentations, des observations de classe, leurs erreurs typiques.
- Enseignants et formateurs : le lexique utilisé, leurs représentations, leurs choix de matériels, leurs parcours dans certaines formations.
- Préconisations : des analyses de programme, de manuels, de ressources, de plateformes mises en place.
- Ressources : des robots, Scratch, plateforme de scénarios incluant des *teaching analytics*. . .

On pourrait inclure d'autres éléments développés dans le projet ANR ne figurant pas dans le livre : raison des choix des professeurs de mathématiques d'enseigner et surtout de ne pas enseigner la programmation, l'étude des représentations des élèves de détecteurs de bruit et des capteurs de CO<sub>2</sub>. . .

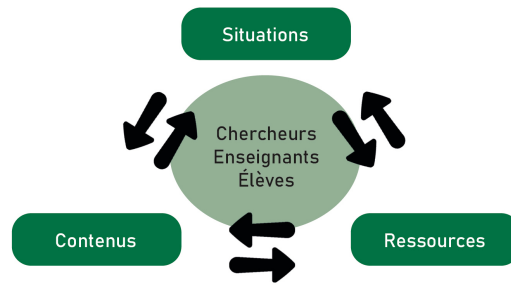
Ainsi, les analyses menées fournissent une idée de la diversité des questions qui peuvent se poser sur l'informatique en éducation, surtout considérée à la fois comme science, technologie et utilisations. Cela peut conduire à se poser la question de la conception d'enseignements scolaires. Le projet ANR IE CARE a constitué en quelque sorte le creuset de la modélisation qui va maintenant être présentée.

## **Le modèle CORESI : contenus, ressources, situations**

Afin de penser une informatique scolaire ouverte, non considérée comme un nouvel avatar d'une discipline ou d'une matière scolaire, nous proposons une modélisation autour de trois pôles : les contenus d'enseignement et d'apprentissage, les situations et les ressources éducatives.

Précisons que cette modélisation est une ébauche qui n'a pas vraiment été investie dans le projet. Il resterait à préciser la sémantique des différents liens ainsi que les manières avec lesquelles les chercheurs, les enseignants et les élèves peuvent être en relation avec les différents pôles. Ainsi, un élève peut proposer une situation (par exemple une situation qu'il a vécue en utilisant une

**FIGURE 2** – Le modèle co(ntenus) re(ssources) si(tuations) (CORESI) (lors de sa première présentation, le modèle, censé mettre de l'ordre dans un environnement chaotique, avait été nommé CRS (contenus, ressources, situations). Cette dénomination, à la tonalité ambiguë et un peu provocatrice, a été modifiée).



application), peut identifier un contenu en posant une question (par exemple en demandant comment fonctionne le détecteur de CO<sub>2</sub> installé dans la classe) ou amener des ressources (par exemple en montrant un traducteur ou un jeu qu'il utilise).

Nous allons explorer la signification que l'on peut attribuer à chacun des pôles et décrire les relations entre eux. Ensuite, nous verrons quelques exemples d'utilisation de ce modèle.

### Pôle contenu

Dans le système éducatif français, les contenus d'enseignement réfèrent d'abord aux programmes prescrits, qui peuvent être précisés dans des référentiels. Selon Fluckiger (2019, p. 52), « la didactique de l'informatique est la science qui identifie des contenus informatiques d'enseignement / apprentissage, les construit en tant qu'objets scientifiques et étudie leurs conditions d'élaboration, de diffusion, de structuration et/ou d'appropriation par les différents acteurs d'un système éducatif ». Cette façon de voir est moins verticale et beaucoup plus dynamique. Elle n'est pas vraiment compatible avec des progressions prédéfinies. Des contenus peuvent être identifiés en cours d'activité, proposés par les uns et les autres. En ce sens, un contenu « relève à la fois du régime des savoirs, du régime des pratiques et du régime des représentations » (*ibid.*, p. 89).

Un contenu est lié à un acte d'enseignement / apprentissage traduisant un savoir vivant, des connaissances vivantes et leurs relations. Bruillard (2021) décrit des exemples caractéristiques liés à l'écriture inclusive sur un site web ou à un correcteur orthographique jugé raciste. L'idée derrière ces exemples est de déterminer ce qu'il faudrait connaître afin de comprendre les dysfonctionnements constatés (de la situation observée aux contenus à identifier).

Des contraintes jugées comme des obstacles peuvent devenir des enjeux et des contenus de formation (dans l'idée d'objectif obstacle de Martinand). Ainsi les contraintes matérielles et technologiques (par exemple le frottement), ouvrent à des contenus

technologiques, les déplacements de robots ne se limitent pas à des éléments de mathématiques élémentaires ni de physique théorique (Grugier, 2022).

On peut également investiguer le fonctionnement des moteurs de recherche, le codage des caractères, les changements de langues, etc., via des détournements, des jeux (Simonnot, 2008) ou faire un cycle de traduction, en revenant à la langue initiale. . .

## **Pôle ressources**

L'informatique et son enseignement conduisent à des questions spécifiques sur les ressources. La plus classique est celle du choix d'un langage de programmation et plus généralement d'un environnement de programmation. Ce choix oriente, voire conditionne les situations éducatives que l'on peut construire et les objectifs et contenus d'enseignement que l'on peut viser et proposer. Cela inclut la construction ou le choix d'artefacts, comme des robots, mais aussi des grilles (dans lesquelles s'organisent des déplacements), etc.

Il y a beaucoup d'exemples concernant les ressources dans cet ouvrage. Le texte sur les types de robots choisis et l'analyse des textes produits par des experts et des enseignants montre leurs relations (chapitre 9 « Connaissances pour enseigner l'informatique : analyse textuelle de productions d'enseignants de l'école primaire »), une plateforme de scénarios (chapitre 11 « Assister les enseignants du primaire dans l'enseignement de la pensée informatique. Une approche basée sur la scénarisation et vers les *teaching analytics* »), une démarche d'une enseignante (chapitre 4 « Quelles ressources pour enseigner l'informatique dans le premier degré? Une étude de cas »). . .

Les ressources peuvent être trouvées dans une offre (celles des robots, des langages, des aides, etc.) dans des collections, des malles, des valises. . . mais aussi produites par les acteurs (enseignants et élèves).

Ce que l'on vient de présenter correspond à des « ressources externes », matérielles ou intellectuelles. On ajoute les « ressources internes », c'est-à-dire les connaissances, compétences, dispositions, des acteurs, que ce soient les enseignants ou les apprenants.

## **Pôle situation**

C'est d'une part ce qui est organisé, mis en place pour faire apprendre aux élèves, ce que l'on fait « vivre » aux élèves. La situation peut comprendre un scénario plus ou moins précis, des

ressources générales à actualiser. Ce sont les tâches prescrites ou les activités menées.

D'autre part, comme on l'a vu, cela peut être des rencontres avec des phénomènes inattendus, que l'on va chercher à expliquer : des énigmes proposées ou non par les enseignants, des détournements plus ou moins provoqués, etc. L'élève peut être seul face à une situation et il essaye de mobiliser des ressources pour la comprendre ou la résoudre. Il peut être accompagné par un enseignant qui identifie les contenus, installe les ressources externes, aide à mobiliser les ressources internes. L'évaluation, quand elle est menée, s'intègre à une situation.

### Quelques relations entre les pôles

Le modèle CORESI, simplement ébauché dans ce texte, associe contenus, ressources et situations, tenant compte des visions des chercheurs, des enseignants et des élèves. Au-delà des relations entre les pôles, il faut penser aux interactions entre les acteurs liés à ces pôles (choix et constructions des contenus, ressources, situations ; assistance dans les résolutions. . .).

Les trois pôles sont intriqués et il est parfois difficile de les séparer. Le choix des ressources conditionne ce qui va être fait, dans le choix même des contenus et des situations. La notion de référence peut se décliner dans les pôles : contenu en référence à une profession / ressource venant du monde professionnel / situation « authentique ».

Des relations entre deux pôles sont également à considérer :

- Contenus et ressources : dialectique outil-objet (Douady, 1986), ce qui est d'abord un objectif d'enseignement devient un outil, une ressource pour d'autres enseignements ; des ressources existantes comme les langages et environnements de programmation, induisent des contenus.
- Contenus et situations : des situations sont construites pour apprendre et enseigner des contenus ; des situations, énigmes, dysfonctionnements à expliquer, conduisent à identifier des contenus.
- Ressources et situations : les ressources permettent de construire des situations et peuvent être mobilisées pour étudier, résoudre des situations. . .

Comme il est classique dans les modélisations ternaires, on peut analyser ce qu'il advient quand un des pôles est minoré ou, au contraire, dominant (comme dans le triangle pédagogique de Houssaye, 2000). Par exemple, en se focalisant sur le pôle contenu, les chercheurs peuvent confronter leurs visions, on peut lister les

grands concepts informatiques (séquence, répétition, structure de contrôle, variables et types, etc.), les pratiques . . .

La conception de plateforme de ressources privilégie les relations ressources-situations (ce que l'on met en place avec quoi), les contenus peuvent être ensuite identifiés dans ce qu'offre la plateforme.

## Exemples d'utilisation du modèle en formation d'enseignants

Le modèle peut faciliter la compréhension de différents types de formation, notamment les formations pour les enseignants en charge d'enseigner de l'informatique. Il s'agit de comprendre sur quoi s'appuie principalement la formation.

Ainsi, une formation que l'on peut qualifier de conceptuelle se centre essentiellement sur les contenus de formation, souvent en association avec des objectifs de formation fixés à l'avance. Il s'agit de sélectionner une liste de concepts puis de construire une formation adaptée pour faire apprendre les différents concepts de cette liste. *A contrario*, une formation dite pratique peut être basée sur un ensemble de situations. Parcourir l'ensemble de ces situations, intégrant différentes ressources, assure plus ou moins l'apprentissage des concepts ou des contenus visés. Dans certains modèles, il s'agit de faire vivre aux enseignants des situations directement utilisables avec les élèves. Ils n'auront plus qu'à les réutiliser, avec ou sans adaptation pour leurs élèves. Une telle approche pose question, puisque les adultes peuvent réagir assez différemment des enfants et cela peut passer sous silence les connaissances et savoirs autres (par exemple sur les mesures de distances, la causalité, etc.) qui sont pourtant essentiels à comprendre dans les difficultés que peuvent rencontrer les enfants.

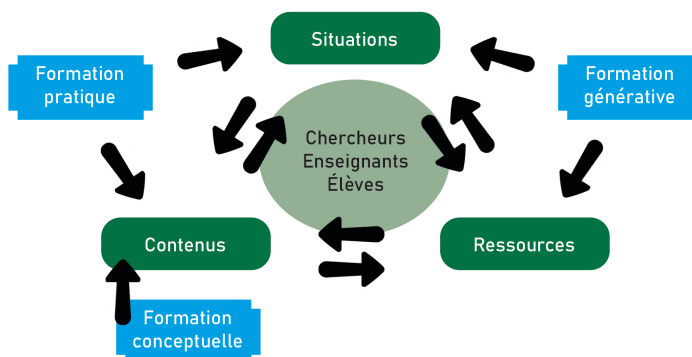


FIGURE 3 – Différents modèles de formation.

3. L'ECOS (examen clinique objectif structuré) est un dispositif d'évaluation des compétences cliniques. Il se présente comme une succession de plusieurs stations qui reconstituent chacune une consultation médicale avec des patientes et patients simulés (PS). L'ensemble des stations par lesquelles passent les candidates et candidats est appelé circuit. Ils et elles y réalisent des tâches cliniques — selon leur niveau d'études — à partir du scénario initial qui leur est proposé : anamnèse, examen physique, diagnostic/diagnostic différentiel, investigations et/ou prise en charge (cf. <https://www.unil.ch/ecoledemedecine/fr/home/menuintst/examens-evaluations/ecos.html>).

Des approches par compétences privilégient les situations, en cohérence avec des définitions des compétences comme des savoir-agir dans des familles de situations (Tardif, 2006, p. 22). Les situations d'apprentissage et d'évaluation (SAE) sont le cœur pour l'organisation des formations dans la réforme des Instituts universitaires de technologie (IUT). On retrouve des préoccupations similaires s'agissant des examens cliniques objectifs structurés (ECOS)<sup>3</sup>.

Le pôle « contenus » a alors tendance à se dissoudre dans le pôle « ressources ».

Une tendance observable vers des formations continues d'enseignants de durée courte conduit à privilégier des modèles de formation pratique : on équipe les enseignants d'un ensemble de situations directement réutilisables en un temps court, sans peut-être leur donner une formation conceptuelle suffisante pour qu'ils puissent être à l'aise dans l'identification des contenus et la construction et l'évaluation des situations.

Il n'y a pas, à proprement parler, de formations basées avant tout sur les ressources, mais certaines d'entre elles, que l'on peut qualifier de génératives, vont se concentrer sur la conception de ressources et de situations. L'apprentissage des concepts est censé avoir été fait ou peut se compléter via la conception des situations.

On peut prendre ce modèle pour concevoir des formations avec des progressions, mais également des formations qui correspondent à des suites de rencontres. Comme dans certaines approches par compétences, le pôle « situation » est dominant, ce sont les rencontres, qui orientent ou déterminent les ressources et l'identification des contenus.

## Du code à la pensée informatique : comment sortir des répétitions

Le début d'IE CARE s'inscrit dans un contexte où l'apprentissage du code ou du codage était revendiqué. Cette idée de code résonne avec le permis de conduire informatique, promu dans les années 90 en Europe sous le nom de *European Computer Driving Licence* (ECDL) attestant les compétences pratiques d'utilisation des applications informatiques les plus courantes (puis PCIE passeport de compétences informatiques européen). Notons que ECDL s'est transformé en *International Certification of Digital Literacy* (ICDL)<sup>4</sup>. Le sigle ECDL est conservé en Suisse sous la dénomination *European Certificate of Digital Literacy*<sup>5</sup>. La nouveauté dans l'éducation nationale française résidait dans la pression pour l'apprentissage de la programmation, une vision très différente de l'appropriation d'applications informatiques courantes.

4. <https://icdleurope.org/fr/>

5. <https://www.ecdl.ch/fr/ecdl/quest-ce-que-lecdl/>

Le rapport de l'Inspection générale (Gaubert-Macon *et al.*, 2022) consacré à la pratique de l'informatique aux cycles 3 et 4, répond à la demande du ministre de l'Éducation d'octobre 2021. Il est notamment question de la « contribution à la pensée informatique ».

Or dans les programmes officiels de 2016<sup>6</sup>, l'expression « pensée informatique » est absente. Ce que relève également le rapport (p. 5) :

Si ce concept est absent des programmes, il fait néanmoins l'objet d'une évaluation dans le cadre de l'enquête *International Computer and Information Literacy Study* (ICILS) pour laquelle un panel d'élèves de quatrième a été testé en 2018 et sera testé en 2023.

6. <https://www.education.gouv.fr/au-bo-special-du-26-novembre-2015-programmes-d-enseignement-de-l-ecole-elementaire-et-du-college-3737>

Il est précisé, pour le cycle 3, « des programmes ne faisant pas explicitement référence à la pensée informatique » et des programmes qualifiés d'« en lien avec la pensée informatique » en cycle 4 en montrent des libellés de compétences que l'on peut rattacher à la dite pensée informatique. Seules deux mentions de la « pensée algorithmique » apparaissent. De plus, le rapport affirme dès la première ligne de sa synthèse que « l'informatique est une science », précisant :

Dans le domaine éducatif le terme « informatique » souffre d'ambiguïté, souvent confondu avec l'usage de solutions logicielles ou encore la littératie numérique.

On en revient à la seule science : quels rapports entre cette science et ladite « pensée informatique » ?

Nous n'irons pas plus avant dans l'analyse du rapport. Il apparaît des écarts ou des contradictions dans la « pensée ministérielle » et on peut se demander s'il y a des différences entre apprentissage de la programmation et pensée informatique dans les discours scolaires. En outre, ces subtilités sont éloignées des représentations des acteurs.

Ainsi, l'expression « pensée informatique » s'associe à l'apprentissage de la programmation, excluant d'autres réflexions. Mais cette pensée, mal définie, se réduit souvent à la programmation et à des progressions liées à son apprentissage, sans grande ouverture.

Le cas de la répétition prédéfinie est caractéristique. Elle est prônée car son emploi par les élèves est censé indiquer leur capacité à identifier des *patterns* et à prévoir, dans une forme d'écriture condensée, leur répétition et le nombre de répétitions à effectuer. La répétition prédéfinie pose des problèmes aux élèves (Touloupaki, 2023). Or, dans diverses situations, il est bien meilleur de copier et de faire plusieurs coller d'une suite d'instructions



ou d'un *pattern*. On peut ajuster au fur et à mesure et modifier des éléments du *pattern* utilisé. Par exemple, on peut répéter la même phrase musicale et ensuite faire des petites modifications dans certaines. Répéter x fois le même *pattern* interdit d'introduire de petites modifications au cours des répétitions, aussi ce que certains peuvent considérer comme des avancées conduit alors à une réduction. Sans s'en douter, sous la bannière de la « pensée informatique », peut s'opérer une forme d'assèchement.

Pour terminer, si des enseignants continuent à inventer des situations intéressantes avec leurs élèves, les formes de scolarisation actuelles apparaissent peu concluantes. Espérons que la créativité des enseignants puisse faire émerger des contenus informatiques et des situations associées, soutenues par des ressources, afin d'offrir des opportunités d'apprentissage intéressantes pour les élèves.

## Références

Baron, G-L. (1989). *L'informatique, discipline scolaire ? : le cas des lycées*, Presses universitaires de France.

Boissière, J. et Bruillard, É. (2021). *L'école digitale. Une éducation à construire et à vivre*, Armand Colin.

Bruillard, É. (2019). Enseigner à l'ère de l'internet : catalogues, couleurs et collections, *Raison présente*, vol. 210, p. 37-49. <https://doi.org/10.3917/rpre.210.0037>

Bruillard, É. (2021). L'écriture inclusive ouvre des liens surprenants. Réflexions en didactique de l'informatique, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 27, n° 1. <https://doi.org/10.23709/sticf.27.1.4>

Douady, R. (1986). Jeux de cadres et dialectique outil-objet, *Recherches en didactique des mathématiques*, vol. 7, n° 2, p. 5-31. <https://revue-rdm.com/1986/jeux-de-cadres-et-dialectique/>

Drot-Delange, B., Pellet, J-P., Delmas-Rigoutsos, Y., et Bruillard, É. (2019). « Pensée informatique : points de vue contrastés », dans *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 26, p. 39-61.

Fluckiger, C. (2019a). *Une approche didactique de l'informatique scolaire*, Presses universitaires de Rennes.

Gaubert Macon, C., Chesneaux, J-M., Desprez, J-M., Picaronny, C., et Montreuil, V. (2022). Pratique de l'informatique aux cycles 3 et 4, *Rapport de l'inspection générale de l'éducation, du sport et de la recherche*, vol. 21-22, n° 169A. <https://www.education.gouv.fr/pratique-de-l-informatique-aux-cycles-3-et-4-344254>

Grugier, O. (2021). Manipulations de robots programmables en classe par des élèves de 9-10 ans. *Éducation au numérique et culture technique, Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 28, n° 3, p. 71-94. <https://hal.science/hal-03671165v1>

Houssaye, J. (2000). *Le triangle pédagogique. Théorie et pratiques de l'éducation scolaire*, Éditions Peter Lang.

Simonnot, B. (2008). Quand les moteurs de recherche appellent au jeu : usages ou détournements ?, *Questions de communication*, vol. 14, p. 95-114. <https://doi.org/10.4000/questionsdecommunication.752>

Tardif, J. (2006). *L'évaluation des compétences. Documenter le parcours de développement*, Chenelière Éducation.

Touloupaki, S. (2023). Contribution à l'étude de l'apprentissage de la programmation en grande section et en cours préparatoire, à travers le logiciel ScratchJr : une approche didactique exploratoire, thèse de l'université Paris Cité. <https://theses.fr/s177034>

Pour citer ce chapitre :

Bruillard, Eric (2024). « De modèles d'analyse à un modèle de conception de formations à l'informatique », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 235-247. <https://doi.org/10.53480/2024iecare0cz>





# **BIBLIOGRAPHIE**



## Références

Abiteboul, S., Archambault, J.-P., Balagué, C., Baron, G.-L., Berry, G., Dowek, G., de la Higuera, C., Nivat, M., Tort, F., et Viéville, T. (2015). *L'enseignement de l'informatique en France – Il est urgent de ne plus attendre*, Académie des sciences, p. 35. <https://www.academie-sciences.fr/fr/Rapports-ouvrages-avis-et-recommandations-de-l-Academie/l-enseignement-de-l-informatique-en-france-il-est-urgent-de-ne-plus-attendre.html>

Abric, J.-C. (2003). L'étude expérimentale des représentations sociales, dans D. Jodelet (dir.), *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 203-223. <https://doi.org/10.3917/puf.jodel.2003.01>

Académie des sciences (2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*, Rapport de l'Académie des sciences. [https://www.academie-sciences.fr/pdf/rapport/rads\\_0513.pdf](https://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf)

Alayrangues, S., Peltier, S., et Signac, L. (2017). Informatique débranchée : construire sa pensée informatique sans ordinateur, *colloque Mathématiques en cycle 3 IREM de Poitiers*, p. 216-226, Poitiers, France. <https://hal.science/hal-01868132>

Alemán de la Garza, L., Bruillard, E., Gómez-Zermeño, M., et al. (2019). *Rethinking pedagogy : Exploring the potential of digital technology in achieving quality education*, UNESCO-UNESDOC Mahatma Gandhi Institute of Education for Peace and Sustainable Development (MGIEP). <https://unesdoc.unesco.org/ark:/48223/pf0000372786>

Allouche, E. (2021). Les humanités numériques, pour un dialogue interdisciplinaire entre recherche et éducation, *Frantice.net*, n° 17. <https://hal.science/hal-03080381>

Amadiou, F., et Tricot, A. (2014). *Apprendre avec le numérique. Mythes et réalités*, Retz. <https://doi.org/10.14375/NP.9782725633206>

Anseel, F., Lievens, F., et Schollaert, E. (2009). Reflection as a strategy to enhance task performance after feedback, *Organizational*

*Behavior and Human Decision Processes*, vol. 110, n° 1, p. 23-35. <https://doi.org/10.1016/j.obhdp.2009.05.003>

Aristote (1840). *Métaphysique, Livre I*. (Victor Cousin trad., texte original publié au IV<sup>e</sup> siècle av. J.-C.). <http://remacle.org/bloodwolf/philosophes/Aristote/metaphysique1fr.htm>

Arleo, A., et Delalande, J. (2010). *Cultures enfantines : universalité et diversité*, Presses universitaires de Rennes. <https://doi.org/10.4000/books.pur.10732>

Bardach, L., Klassen, R. M., Durksen, T. L., Rushby, J. V., Bostwick, K. C. P., et Sheridan, L. (2021). The power of feedback and reflection : Testing an online scenario-based learning intervention for student teachers, *Computers & Education*, vol. 169, p. 104-194. <https://doi.org/10.1016/j.compedu.2021.104194>

Bardin, L. (2013). *L'analyse de contenu*, Presses universitaires de France. <https://doi.org/10.3917/puf.bard.2013.01>

Baron, G-L. (1989). *L'informatique, discipline scolaire ? : le cas des lycées*, Presses universitaires de France.

Baron, G.-L. (2018). *Informatique et numérique comme objets d'enseignement scolaire en France : entre concepts, techniques, outils et culture*, Didapro 7 – DidaSTIC. De 0 à 1 ou l'heure de l'informatique à l'école. <https://hal.archives-ouvertes.fr/hal-01760479/document>

Baron, G-L. (2021). Brèves réflexions sur les humanités numériques, *Frantice.net*, n° 17. <https://hal.science/hal-03080381>

Baron, G.-L., et Boulc'h, L. (2011). Les technologies de l'information et de la communication à l'école primaire. État de question en 2011, *Revue de l'enseignement public et informatique (EPI)*. <https://www.epi.asso.fr/revue/articles/a0910c.htm>

Baron, G.-L., et Bruillard, É. (1996). *L'informatique et ses usagers dans l'éducation*, PUF. [http://mutatice.net/glbaron/lib/exe/fetch.php/baron\\_bruillard\\_livre1996-fprepub2.pdf](http://mutatice.net/glbaron/lib/exe/fetch.php/baron_bruillard_livre1996-fprepub2.pdf)

Baron, G.-L., et Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie. Recherches en éducation*, vol. 195, n° 2, p. 51-62. <https://doi.org/10.4000/rfp.5032>

Baron, G.-L., et Fluckiger, C. (2021). Approches et paradigmes pour la recherche sur les usages éducatifs des technologies : enjeux et perspectives, *Canadian Journal of Learning and Technology*, vol. 47, n° 4. <https://doi.org/10.21432/cjlt28059>

Baron, G.-L., Drot-Delange, B., Grandbastien, M., et Tort, F. (2014). Computer science education in French secondary schools : Historical and didactical perspectives, *Trans. Comput. Educ.*, vol. 14, n° 2, p. 1-27. <https://doi.org/10.1145/2602486>

- Baron, G.-L., Bruillard, E., et Drot-Delange, B. (2015). *Informatique en éducation : perspectives curriculaires et didactiques*, Presses universitaires Blaise-Pascal. <https://hal.science/hal-01136337>
- Bart, D., et Fluckiger, C. (2015). « Évaluation, fabrication des contenus et disciplines d'enseignement », dans B. Daunay, C. Fluckiger, et R. Hassan, *Les Contenus d'enseignement et d'apprentissage. Approches didactiques*, Presses universitaires de Bordeaux, p. 91-102. <https://doi.org/10.4000/books.pub.38357>
- Baudé, J. (2018). *Le séminaire de Sèvres (mars 1970)*, Revue de l'EPI, n° 201.
- Bautier, É. (2002). Du rapport au langage : Question d'apprentissages différenciés ou de didactique? *Pratiques*, vol. 113, n° 1, p. 41-54. <https://doi.org/10.3406/prati.2002.1944>
- Bautier, E., et Goigoux, R. (2004). Difficultés d'apprentissage, processus de secondarisation et pratiques enseignantes : une hypothèse relationnelle, *Revue française de pédagogie*, vol. 148, n° 1, p. 89-100. <https://doi.org/10.3406/rfp.2004.3252>
- Beaudouin, V. (2016). Retour aux origines de la statistique textuelle : Benzécri et l'école française d'analyse des données, *Journées internationales d'Analyse statistique des Données Textuelle*, Nice, France, p. 17-27. <https://hal.science/hal-01376938>
- Beillerot, J. (1989). « Le rapport au savoir », dans J. Beillerot, A. Bouillet, C. Blanchard-Laville. N. Mosconi et P. Obertelli (dir.), *Savoir et rapport au savoir. Elaboration théorique et clinique*, Éditions universitaires, p. 165-202.
- Bellegarde, K. (2015). *Littéracie et appropriation : ruptures et continuités dans les rapports à l'écrit d'adultes en parcours d'insertion / formation*, thèse de l'université Lille 1. <https://www.theses.fr/2015LIL12012>
- Bellegarde, K., Boyaval, J., et Alvarez, J. (2021). Initier des élèves de maternelle à la robotique/informatique : quand les supports médiateurs impactent la grammaire de l'agir enseignant, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 28, n° 3, p. 13-38. <https://doi.org/10.23709/sticf.28.3.5>
- Benitti, F.B.V. (2012). Exploring the educational potential of robotics in schools : A systematic review, *Computers and Education*, vol. 58, n° 3, p. 978-988. <https://doi.org/10.1016/j.compedu.2011.10.006>
- Bennacer, I. (2022). Teaching analytics : Support for the evaluation and assistance in the design of teaching through artificial intelligence, thèse de l'université du Mans. <https://theses.hal.science/tel-03935709>



- Benzécri, J. P. (1973). *L'analyse des données, Tome 1 : La Taxinomie; Tome 2 : L'analyse des correspondances*, Dunod.
- Bers, M. U. (2010). The tangible K robotics program : Applied computational thinking for young children, *Early Childhood Research & Practice*, vol. 12, n° 2. <http://eric.ed.gov/?id=EJ910910>
- Bers, M.U., Flannery, L., Kazakoff, E.R., et Sullivan, A. (2014). Computational thinking and tinkering : Exploration of an early childhood robotics curriculum, *Computers and Education*, vol. 72, p. 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Béziat, J. (2012). Informatique, outil ou objet? Permanence d'une question. Le cas de l'école primaire en France, *Adjectif : analyses et recherches sur les TICE*, vol. 3. <http://www.adjectif.net/spip/spip.php?article177>
- Béziat, J. (2019). À l'école primaire, robotique éducative en milieu ordinaire, *Spirale - revue de recherche en éducation*, vol. 63. <https://doi.org/10.3917/spir.063.0091>
- Bisault, J., Le Bourgeois, R., Thémines, J.F., Le Mentec, M., et Chauvet-Chanoine, C. (2023). *Objets pour apprendre, objets à apprendre. Quelles pratiques enseignantes pour quels enjeux?*, ISTE group.
- Blaya, C., et Alava, S. (2012). *Risques et sécurité des enfants sur Internet : rapport pour la France-résultats de l'enquête EU Kids Online menée auprès des 9-16 ans et de leurs parents en France*. <http://eprints.lse.ac.uk/46443/1/FranceReportFrench.pdf>
- Blin, J.-F. (1997). *Représentations, pratiques et identités professionnelles*, L'Harmattan.
- Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., et Franklin, D. (2013). Hairball : Lint-inspired static analysis of scratch projects, *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013)*, p. 215-220. <https://doi.org/10.1145/2445196.2445265>
- Boissière, J. et Bruillard, É. (2021). *L'école digitale. Une éducation à construire et à vivre*, Armand Colin.
- Bonnardel, N., et Lubart, T. (2019). La créativité : approches et méthodes en psychologie et en ergonomie, *RIMHE : Revue interdisciplinaire management, homme et entreprise*, vol. 37, n° 8, p. 79-98. <https://doi.org/10.3917/rimhe.037.0079>
- Bourmaud, G., et Rabardel, P. (2005). « Instruments et systèmes d'instruments », dans P. Rabardel, P. Pastré (dir.), *Modèles du sujet pour la conception. Dialectiques activités développement*, Octarès.
- Braitenberg, V. (1984). *Vehicles : Experiments in synthetic psychology*, MIT Press.

- Brennan, K., et Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA 2012)*, vol. 1, n° 25. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Brinkerhoff, J. (2006). Effects of a long-duration, professional development academy on technology skills, computer self-efficacy, and technology integration beliefs and practices, *Journal of Research on Technology in Education*, vol. 39, n° 1, p. 22-43. <https://doi.org/10.1080/15391523.2006.10782471>
- Bronfenbrenner, U. (1979). *The Ecology of Human Development : Experiments by Nature and Design*, Harvard University Press. <https://doi.org/10.4159/9780674028845>
- Brown, A. L., Lee, J., et Collins, D. (2015). Does student teaching matter? Investigating pre-service teachers' sense of efficacy and preparedness, *Teaching Education*, vol. 26, n° 1, p. 77-93. <https://doi.org/10.1080/10476210.2014.95766>
- Bruillard, É. (2009a). « Place de l'informatique dans l'enseignement secondaire, réflexions introductives », dans G.-L. Baron, É. Bruillard, et L.-O. Pochon (dir.), *Informatique et progiciels en éducation et en formation*, p. 21-27, INRP.
- Bruillard, E. (2009b). Place de l'informatique dans l'enseignement secondaire, *EpiNet : la revue électronique de l'EPI*. <https://edutice.archives-ouvertes.fr/edutice-00558936/file/a0910c.htm>
- Bruillard, E. (2010). « Acteurs et territoires de l'éducation à l'information : un point de vue 'informatique' », dans Françoise Chapron et Éric Delamotte (dir.), *L'éducation à la culture informationnelle*, Presses de l'Ensib, p. 68-75. <https://doi.org/10.4000/books.pressesensib.837>
- Bruillard, É. (2014). *Une voie pour penser et construire une formation à l'informatique pour les élèves de l'école primaire ?* <https://hal.science/hal-03948939>
- Bruillard, É. (2019). Enseigner à l'ère de l'internet : catalogues, couleurs et collections, *Raison présente*, vol. 210, p. 37-49. <https://doi.org/10.3917/rpre.210.0037>
- Bruillard, É. (2021). L'écriture inclusive ouvre des liens surprenants. Réflexions en didactique de l'informatique, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 27, n° 1. <https://doi.org/10.23709/sticf.27.1.4>
- Bruillard, É., et Fluckiger, C. (2018). Une approche didactique de l'informatique scolaire, *Revue française de pédagogie, recherche en éducation*, vol. 204, p. 86-87. <https://doi.org/10.4000/rfp.8523>

Brunet, O., Yessad, A., Muratet, M., et Carron, T. (2020). Vers un modèle de scénarisation pour l'enseignement de la pensée informatique à l'école primaire, *Didapro 8 – DidaSTIC*. <https://hal.science/hal-02496191>

Bulletin officiel (2015). Socle commun de connaissances, de compétences et de culture, BO n° 17 du 23 avril 2015. <https://www.education.gouv.fr/bo/15/Hebdo17/MENE1506516D.htm>

Cardon, D. (2019). « Introduction. Coder, décoder », dans : Dominique Cardon (dir.), *Culture numérique*, Presses de Sciences Po, p. 5-13. <https://www-cairn-info.ezproxy.u-paris.fr/culture-numerique-9782724623659-page-5.htm>

Carlson, J., Daehler, K. R., Alonzo, A. C., Barendsen, E., Berry, A., Borowski, A., Carpendale, J., Kam Ho Chan, K., Cooper, R., Friedrichsen, P., Gess-Newsome, J., Henze-Rietveld, I., Hume, A., Kirschner, S., Liepertz, S., Loughran, J., Mavhunga, E., Neumann, K., Nilsson, P., (. . .) Wilson, C. D. (2019). « The refined consensus model of pedagogical content knowledge in science education », dans A. Hume, R. Cooper, et A. Borowski (dir.), *Repositioning Pedagogical Content Knowledge in Teachers' Knowledge for Teaching Science*, Springer, p. 77-94. [https://doi.org/10.1007/978-981-13-5898-2\\_2](https://doi.org/10.1007/978-981-13-5898-2_2)

Carré, P. (1998). Motifs et dynamiques d'engagement en formation. Synthèse d'une étude qualitative de validation auprès de 61 adultes en formation professionnelle continue, *Éducation permanente*, vol. 136, p. 119-131.

Carré, P., et Caspar, P. (2017). *Traité des sciences et des techniques de la Formation*, 4<sup>e</sup> éd., Dunod. <https://doi.org/10.3917/dunod.carre.2017.01>

Charlot, B. (1997). *Du rapport au savoir. Éléments pour une théorie*, Anthropos.

Charlot, B. (1999). *Le rapport au savoir en milieu populaire*, Anthropos. <https://doi.org/10.4000/books.editions-cnrs.31266>

Charlot, B. (2001). *Les jeunes et le savoir. Perspectives internationales*, Anthropos.

Chen, G. (2020). A visual learning analytics (VLA) approach to video-based teacher professional development : Impact on teachers' beliefs, self-efficacy, and classroom talk practice, *Computers & Education*, vol. 144, p. 103-670. <https://doi.org/10.1016/j.compedu.2019.103670>

Chervel, A. (1988). L'histoire des disciplines scolaires. Réflexions sur un domaine de recherche, *Histoire de l'éducation*, vol. 38, n° 1, p. 59-119. <https://doi.org/10.3406/hedu.1988.1593>

Chevallard, Y. (1985). *La transposition didactique. Du savoir savant au savoir enseigné*. Le Pensée Sauvage.

Cibois, P. (1990). Éclairer le vocabulaire des questions ouvertes par les questions fermées : Le tableau lexical des questions, *Bulletin of Sociological Methodology / Bulletin de Méthodologie Sociologique*, vol. 26, n° 1, p. 12-21. <https://doi.org/10.1177/075910639002600102>

Clot, Y., et Faïta, D. (2000). Genres et styles en analyse de travail. Concepts et méthodes, *Travailler*, vol. 4, p. 7-42.

Combarrous, M. (1984). *Comprendre les techniques et la technicité*, Éditions sociales.

Couderette, M., Haspekian, M., et Kiwan, M. (2023). « Algorithmique, programmation, informatique », dans *Les prescriptions en mathématiques : un premier repérage* [présentation orale], Séminaire IE CARE, Paris, France.

Crahay, M. (1987). Logo, un environnement propice à la pensée procédurale, *Revue française de pédagogie*, vol. 80, n° 1, p. 37-56. <https://doi.org/10.3406/rfp.1987.1473>

Crédoc (2021). *Baromètre du numérique, édition 2021*. <https://www.credoc.fr/publications/barometre-du-numerique-edition-2021>

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.researchgate.net/publication/327302966\\_Computational\\_thinking\\_-\\_a\\_guide\\_for\\_teachers](https://www.researchgate.net/publication/327302966_Computational_thinking_-_a_guide_for_teachers)

Daunay, B., et Reuter, Y. (2013). « Penser et problématiser les contenus disciplinaires : un enjeu fondamental pour les didactiques », dans B. Daunay, Y. Reuter et A. Thépaut (dir.), *Les contenus disciplinaires. Approches comparatistes*, p. 21-34, Presses universitaires du Septentrion. <https://doi.org/10.4000/books.septentrion.15740>

Daunay, B., Fluckiger, C., et Hassan, R. (2015). *Les contenus d'enseignement et d'apprentissage. Approches didactiques*. Presses universitaires de Bordeaux. <https://doi.org/10.4000/books.pub.38242>

Dayer, C., et M. Charmillot (2012). Démarche compréhensive et méthodes qualitatives : clarifications épistémologiques, *Formations et pratiques d'enseignement en questions*, vol. 14, p. 163-176

De Coninck, K., Valcke, M., Ophalvens, I., et Vanderlinde, R. (2019). « Bridging the theory-practice gap in teacher education : The design and construction of simulation-based learning environments », dans K. Hellmann, J. Kreutz, M. Schwichow, et K. Zaki (dir.), *Kohärenz in der Lehrerbildung*, Springer Fachmedien Wiesbaden, p. 263-280. [https://doi.org/10.1007/978-3-658-23940-4\\_17](https://doi.org/10.1007/978-3-658-23940-4_17)

- Delalande, J. (2003). Culture enfantine et règles de vie, *Terrain. Anthropologie et sciences humaines*, vol. 40, p. 99-114. <https://doi.org/10.4000/terrain.1555>
- Delcambre, I. (2013). « Contenus d'enseignement et d'apprentissage », dans Y. Reuter (dir.), *Dictionnaire des concepts fondamentaux des didactiques*, 3<sup>e</sup> éd., De Boeck, p. 43-48. <https://doi.org/10.3917/dbu.reute.2013.01.0043>
- Denning, P. J. (2017). Remaining trouble spots with computational thinking, *Communications of the ACM*, vol. 60, n° 6, p. 33-39. <https://doi.org/10.1145/2998438>
- Dillenbourg, P. (2015). *Orchestration graphs : Modeling scalable education*, EPFL press.
- Dorais, L.-J. (2004). « La construction de l'identité », dans Denise Deshaies et Diane Vincent (dir.), *Discours et constructions identitaires*, Presses de l'université de Laval.
- Douady, R. (1986). Jeux de cadres et dialectique outil-objet, *Recherches en didactique des mathématiques*, vol. 7, n° 2, p. 5-31. <https://revue-rdm.com/1986/jeux-de-cadres-et-dialectique/>
- Dowek, G. (2011). Les quatre concepts de l'informatique, *Sciences et technologies de l'information et de la communication en milieu éducatif*, EduTice, p. 21-29. <https://edutice.archives-ouvertes.fr/edutice-00676169>
- Dowek, G. (2012). « Les quatre concepts de l'informatique », dans G.-L. Baron, É. Bruillard, et V. Komis, *Actes du quatrième colloque international DIDAPRO 4—Dida&Stic*, New Technologies Editions, p. 21-29.
- Drot-Delange, B., et Fluckiger, C. (2022). « Dialogue sur la didactique de l'informatique : Bilan, enjeux et perspectives », dans Éric Delamotte (dir.), *Recherches francophones sur les éducations aux médias, à l'information et au numérique : points de vue et dialogues*, Presses de l'Enssib, p. 216-235. <https://doi.org/10.4000/books.pressese.nssib.17229>
- Drot-Delange, B., Pellet, J-P., Delmas-Rigoutsos, Y., et Bruillard, É. (2019). « Pensée informatique : points de vue contrastés », dans *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 26, p. 39-61.
- Drot-Delange, B., Parriaux, G., et Reffay, C. (2021). Futurs enseignants de l'école primaire : connaissances des stratégies d'enseignement, curriculaires et disciplinaires pour l'enseignement de la programmation, *Recherches en didactique des sciences et des technologies*, vol. 23, p. 55-76. <https://doi.org/10.4000/rdst.3685>
- Durkheim, E. (1987). *Les règles de la méthode sociologique*, PUF.

- Duval, R. (2017). *Understanding the mathematical way of thinking—The registers of semiotic representations*, Springer. <https://doi.org/10.1007/978-3-319-56910-9>
- Effenberger, T., et Pelánek, R. (2022). Design and analysis of microworlds and puzzles for block-based programming, *Computer Science Education*, vol. 32, n° 1, p. 66-104. <https://doi.org/10.1080/08993408.2020.1832813>
- Fernandez, C., Freitas, J. A., Lopes, R. d. D., et Blikstein, P. (2022). *Using video analysis and learning analytics to understand programming trajectories in data science activities with Scratch*, IDC '22 : Proceedings of the 21st Annual ACM Interaction Design and Children, Braga, Portugal, p. 253-260. <https://doi.org/10.1145/3501712.3529742>
- Feurzeig, W., et Papert, S. (1968). « Programming-languages as a conceptual framework for teaching mathematics » dans OTAN, *Programmé / tendances actuelles, Actes d'un colloque OTAN*, p. 233-246.
- Fluckiger, C. (2008). L'école à l'épreuve de la culture numérique des élèves, *Revue française de pédagogie. Recherches en éducation*, vol. 163, p. 51-61. <https://doi.org/10.4000/rfp.978>
- Fluckiger, C. (2011). La didactique de l'informatique et les constructions sociales de la figure des jeunes utilisateurs, *Recherches en Didactiques*, vol. 11, p. 67-84. <https://doi.org/10.3917/rdid.011.0067>
- Fluckiger, C. (2017). À la convergence entre info-doc et informatique/numérique : quels contenus, quel enseignement? *Journée départementale de l'Association des professeurs documentalistes de l'académie de Nantes du 4 janvier 2017*.
- Fluckiger, C. (2019a). *Une approche didactique de l'informatique scolaire*, Presses universitaires de Rennes.
- Fluckiger, C. (2019b). Numérique en formation : des mythes aux approches critiques, *Éducation permanente*, vol. 219, n° 2, p. 19-30. <https://doi.org/10.3917/edpe.219.0019>
- Fluckiger, C., et Bart, D. (2012). L'introduction du B2i à l'école primaire : évaluer des compétences hors d'une discipline d'enseignement?, *Questions Vives*, vol. 7, n° 17, p. 71-87. <https://doi.org/10.4000/questionsvives.1006>
- Fluckiger, C., et Reuter, Y. (2014). Les contenus « informatiques » et leur(s) reconstruction(s) par des élèves de CM2. Étude didactique, *Recherches en éducation*, vol. 18, p. 64-78. <https://doi.org/10.4000/ree.8510>
- Fluckiger, C., Haspekian, M., et Grugier, O. (2021). *Enseigner l'informatique à l'école primaire : comment les enseignants voient les sujets didactiques*, Colloque eTIC 4, Caen, France.

Gaubert Macon, C., Chesneaux, J.-M., Desprez, J.-M., Picaronny, C., et Montreuil, V. (2022). Pratique de l'informatique aux cycles 3 et 4, *Rapport de l'inspection générale de l'éducation, du sport et de la recherche*, vol. 21-22, n° 169A. <https://www.education.gouv.fr/pratique-de-l-informatique-aux-cycles-3-et-4-344254>

Gelman, R., et Gallistel, C. R. (1978). *The child's understanding of number*, Harvard University Press.

Goggins, S. (2012). Group informatics : A multi-domain perspective on the development of teaching analytics, *CEUR Workshop Proceedings*, vol. 894.

Gosselin, P., et Le Coguiec, E. (2006). *La recherche création, Pour une compréhension de la recherche en pratique artistique*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18ph3x1>

Greff, É. (1998). Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants, *Revue Sciences et techniques éducatives*. <https://doi.org/10.3406/stice.1998.1372>

Greff, E. (1999). En quoi le robot Algor constitue-t-il un objet didactique original ?, *Revue de l'enseignement public et informatique*, vol. 93, p. 127-150.

Gronier, G., et Baudet, A. (2021). Psychometric evaluation of the F-SUS : Creation and validation of the French version of the system usability scale, *International Journal of Human-Computer Interaction*, vol. 37, n° 16, p. 1571-1582. <https://doi.org/10.1080/10447318.2021.1898828>

Grugier, O. (2020). Éducation technologique dans des classes de maternelle. Apprentissages premiers dans l'utilisation et la compréhension d'un artefact robotisé, *Recherche en didactique des sciences et des technologies*, vol. 22, p. 61-92. <https://doi.org/10.4000/rdst.3328>

Grugier, O. (2021). Manipulations de robots programmables en classe par des élèves de 9-10 ans. Éducation au numérique et culture technique, *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, vol. 28, n° 3, p. 71-94. <https://hal.science/hal-03671165v1>

Grugier, O., et Nogry, S. (2022). « Professional testimony : a programmable object for learning computer science at elementary school », dans J. Bisault, R. Le Bourgeois, J.-F. Thémines, M. Le Mentec et C. Chauvet-Chanoine. (dir.), *Objects to learn about and objects for learning : which practices for which issues ?*, ISTE group, p. 159-174. <https://doi.org/10.1002/9781119902171.ch8>

Grugier, O., Fluckiger, C., et Haspekian, M. (2021). Enseignement de l'informatique à l'école : choix des artefacts par les



enseignants, Colloque *L'école Primaire au 21<sup>e</sup> siècle*, Cergy, France.  
<https://hal.science/hal-03378486v1>

Guillemette, F., et Luckerhoff, J. (2012). *L'induction en méthodologie de la théorisation enracinée (MTE). Fondements, procédures et usages*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18pgxtm>

Hadopi (2017). *Les 8-14 ans : l'émergence d'une génération de « smartphone natives »*, Hadopi. <https://www.hadopi.fr/actualites/les-8-14-ans-lemergence-dune-generation-de-smartphone-natives>

Harlé, I. (2016). Analyse de reconfigurations disciplinaires : les apports de la didactique à la sociologie, *Spirale*, vol. 58, p. 23-34.  
<https://doi.org/10.3917/spir.058.0023>

Harrari, M. (2000). *Informatique et enseignement élémentaire 1975-1996. Contribution à l'étude des enjeux et des acteurs*, thèse de l'université René Descartes. <https://tel.archives-ouvertes.fr/edutice-00000406/document>

Haspekian, M. (2011). « The co-construction of a mathematical and a didactical instrument », dans M. Pytak, E. Swoboda, et T. Rowland (dir.), *Proceedings of the Seventh congress of the European Society for Research in Mathematics Education*, p. 2298-2307.  
<https://hal.science/hal-01273866v1>

Holden, H., et Rada, R. (2011). Understanding the influence of perceived usability and technology self-efficacy on teachers' technology acceptance, *Journal of Research on Technology in Education*, vol. 43, n° 4, p. 343-367. <https://doi.org/10.1080/15391523.2011.10782576>

Houssaye, J. (2000). *Le triangle pédagogique. Théorie et pratiques de l'éducation scolaire*, Éditions Peter Lang.

Hursen, C., et Fasli, F. G. (2017). Investigating the efficiency of scenario based learning and reflective learning approaches in teacher education, *European Journal of Contemporary Education*, vol. 6, n° 2, p. 264-279. <https://doi.org/10.13187/ejced.2017.2.264>

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., et Rivers, K. (2015). Educational data mining and learning analytics in programming : Literature review and case studies, *Proceedings of the 2015 Innovation and Technology in Computer Science Education (ITiCSE) on Working Group Reports*, p. 41-63. <https://doi.org/10.1145/2858796.2858798>

INRP (1981). *Dix ans d'informatique dans l'enseignement secondaire. 1970-1980*, Institut national de recherche pédagogique.  
[https://www.epi.asso.fr/blocnote/Dix\\_ans\\_INRP\\_1981.pdf](https://www.epi.asso.fr/blocnote/Dix_ans_INRP_1981.pdf)



Institut de France, Académie des sciences (2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*, Académie des sciences. [http://www.academie-sciences.fr/pdf/rapport/rads\\_0513.pdf](http://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf)

Ipsos (2017). *Junior Connect' 2017 : les jeunes ont toujours une vie derrière les écrans !* <https://www.ipsos.com/fr-fr/junior-connect-2017-les-jeunes-ont-toujours-une-vie-derriere-les-ecrans>

Jiang, B., Zhao, W., Zhang, N., et Qiu, F. (2022). Programming trajectories analytics in block-based programming language learning, *Interactive Learning Environments*, vol. 30, n° 1, p. 113-126. <https://doi.org/10.1080/10494820.2019.1643741>

Jodelet, D. (2003). « Présentation de l'ouvrage », dans *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 5-7. <https://doi.org/10.3917/puf.jodel.2003.01.0005>

Jonnaert, P. (2017). La notion de compétence : une réflexion toujours inachevée, *Éthique publique*, vol. 19 n° 1. <https://doi.org/10.4000/ethiquepublique.2932>

Kanaan, M., et Yessad, A. (2023). « A Two-Step Process for Analysing Teacher's Behaviors Using a Scenario-Based Platform », dans C. S. González-González, B. Fernández-Manjón, F. Li, F. J. García-Peñalvo, F. Sciarone, M. Spaniol, A. García-Holgado, M. Area-Moreira, M. Hemmje, et T. Hao (dir.), *Learning Technologies and Systems*, Springer International Publishing, vol. 13 869, p. 53-62. [https://doi.org/10.1007/978-3-031-33023-0\\_5](https://doi.org/10.1007/978-3-031-33023-0_5)

Kazakoff, E.R., Sullivan, A., et Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, vol. 41, n° 4, p. 245-255. <https://doi.org/10.1007/s10643-012-0554-5>

Kermen, I., et Izquierdo-Aymerich, M. (2017). Connaissances professionnelles didactiques des enseignants de sciences : un thème de recherche encore récent dans les recherches francophones, *Recherches en didactique des sciences et des technologies*, vol. 15, p. 9-32. <https://doi.org/10.4000/rdst.1479>

Khan, M., et Khan, S. S. (2011). Data and information visualization methods, and interactive mechanisms : A survey, *International Journal of Computer Applications*, vol. 34, n° 1, p. 1-14.

Klassen, R. M., Rushby, J. V., Maxwell, L., Durksen, T. L., Sheridan, L., et Bardach, L. (2021). The development and testing of an online scenario-based learning activity to prepare preservice teachers for teaching placements, *Teaching and Teacher Education*, vol. 104, p. 103-385. <https://doi.org/10.1016/j.tate.2021.103385>

Komis, V., et Misirli, A. (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle : une étude

de cas basée sur le jouet programmable Bee-Bot, *Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques*, vol. 271. <https://edutice.hal.science/edutice-00676143>

Komis, V., et Misirli, A. (2012). Jeux programmables de type Logo à l'école maternelle, *Adjectif : analyse et recherche sur les TICE*. <http://www.adjectif.net/spip/spip.php?article140&lang=fr>

Komis, V., et Misirli A. (2015). « Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables », dans Drot-Delange B., Baron G.-L. et Bruillard E. (dir.), *Informatique en éducation : perspectives curriculaires et didactiques*. Presses universitaires Blaise-Pascal, p. 209-226.

Komis, V., Tzavara, A., Karsenti, T., Collin, S., et Simard, S. (2013). Educational scenarios with ICT : An operational design and implementation framework, *Society for Information Technology & Teacher Education International Conference*, p. 3244-3251. [https://www.researchgate.net/publication/289988466\\_Educational\\_scenarios\\_with\\_ICT\\_an\\_operational\\_design\\_and\\_implementation\\_framework](https://www.researchgate.net/publication/289988466_Educational_scenarios_with_ICT_an_operational_design_and_implementation_framework)

Komis, V., Romero, M., et Misirli, A. (2017). « A scenario-based approach for designing educational robotics activities for co-creative problem solving », dans D. Alimisis, M. Moro, et E. Menegatti (dir.), *Educational Robotics in the Makers Era*, Springer, p. 158-169. <https://doi.org/10.1007/s10643-012-0554-5>

Kradolfer, S., Dubois, S., Riedo, F., Mondada, F., et Fassa, F. (2014). « A Sociological contribution to understanding the use of robots in schools : The thymio robot », dans M. Beetz, B. Johnston, et M.-A. Williams (dir.), *Social Robotics*, Springer International Publishing, p. 217-228. [https://doi.org/10.1007/978-3-319-11973-1\\_22](https://doi.org/10.1007/978-3-319-11973-1_22)

Ku, O., Liang, J.-K., Chang, S.-B., et Wu, M. (2018). Sokrates Teaching Analytics System (STAS) : An Automatic Teaching Behavior Analysis System for Facilitating Teacher Professional Development, *26th International Conference on Computers in Education*, Metro Manila, Philippines.

Lahire, B. (1993). *La raison des plus faibles : Rapport au travail, écritures domestiques et lectures en milieux populaires*, FeniXX.

Lahire, B. (2008). La forme scolaire dans tous ses états, *Swiss Journal of Educational Research*, vol. 30, n° 2, p. 229-258. <https://doi.org/10.24452/sjer.30.2.4790>

Lavigne, M. (2023). Formes et sens de l'innovation éducative gamifiée. Une étude de cas : La plateforme Pix, *Technologie et innovation*, vol. 8, n° 3. <https://doi.org/10.21494/ISTE.OP.2023.0972>

Le Touzé, J.-C., N'gosso, I., Robert, F., et Salamé, N. (1979). *Apports d'un environnement informatique dans le processus d'apprentissage*.

*Projet LOGO*, Institut national de recherche pédagogique. Département de recherche sur les applications éducatives des technologies de communication, section « informatique et enseignement ».

Lebart, L., Pincemin, B., et Poudat, C. (2019). *Analyse des données textuelles*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctvq4bxws>

Lebeaume, J. (2019). Objets puis systèmes techniques au programme : éclairages pour une discussion de leurs statuts et de leurs fonctions dans l'enseignement, *Recherches en didactiques*, vol. 27, p. 13-24. <https://doi.org/10.3917/rdid.027.0011>

Leclercq, V., Bellegarde, K. (2015). « Appropriation de l'écrit par des adultes faibles lecteurs et faibles scripteurs : rôle des conceptions de la langue et de son apprentissage », dans *Actes du colloque opale, pour un Français convivial : s'appropriier la langue*, De Boeck, p. 129-144. <https://doi.org/10.3917/dbu.slfc1.2015.01.0129>

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., et Werner, L. (2011). Computational thinking for youth in practice, *ACM Inroads*, vol. 2, n° 1, p. 32-37. <https://doi.org/10.1145/1929887.1929902>

Lefeuvre, G., Garcia A. et Namolovan, L. (2009). Les indicateurs de développement professionnel, *Questions Vives*, vol. 5, n° 11. <https://doi.org/10.4000/questionsvives.627>

Légifrance (2013). *Loi d'orientation et de programmation pour la refondation de l'école de la république (8 juillet 2013)*. <https://www.legifrance.gouv.fr/dossierlegislatif/JORFDOLE000026973437/>

Léonard, M., Peter, Y., Secq, Y., et Fluckiger, C. (2022a). Computational Thinking : Focus on Pattern Identification, *Lecture Notes in Computer Science*, vol. 13 450, p. 187-200. [https://doi.org/10.1007/978-3-031-16290-9\\_14](https://doi.org/10.1007/978-3-031-16290-9_14)

Léonard, M., Secq, Y., Peter, Y., et Fluckiger, C. (2022b). « Pensée informatique : approche didactique de l'identification de motifs », dans *L'informatique, objets d'enseignement et d'apprentissage. Quelles nouvelles perspectives pour la recherche?*, p. 113-125. <https://hal.science/hal-03697888/>

Léonard, M., Bouton, M., et Peter, Y. (2023). Détermination de profils relatifs à la mobilisation de schème lors de la résolution de puzzles de programmation, *Actes de la onzième conférence sur les environnements informatiques pour l'apprentissage humain*, Brest, France.

Levy, S. T., et Mioduser, D. (2008). Does it "want" or "was it programmed to..."? Kindergarten children's explanations of

an autonomous robot's adaptive functioning, *International Journal of Technology and Design Education*, vol. 18, n° 4, p. 337-359. <https://doi.org/10.1007/s10798-007-9032-6>

Livingstone, S. (2012). Critical reflections on the benefits of ICT in education, *Oxford review of education*, vol. 1, n° 38, p. 9-24. <https://doi.org/10.1080/03054985.2011.577938>

Longuet, F. (2012). *L'impact des outils d'évaluation qualitative et du WEB 2.0 sur le développement et l'identification des compétences professionnelles des enseignants de langues*, thèse de l'université Sorbonne Nouvelle - Paris III. <http://tel.archives-ouvertes.fr/tel-00770640/>

Longuet, F. (2014). Former les enseignants de Français langue étrangère par l'activité de création numérique dialoguée, *Synergies Espagne*, vol. 7, p. 189-204. <http://gerflint.fr/Base/Espagne7/Longuet.pdf>

Longuet, F. (2018). Créativité, dialogisme et multimodalité : trois qualités d'un environnement de conception numérique collaboratif, *Revue internationale de pédagogie de l'enseignement supérieur*, vol. 34, n° 2. <https://doi.org/10.4000/ripes.1450>

Longuet, F., et Springer, C. (2021). *Autour du CECR Volume Complémentaire (2018) : Médiation et collaboration. Une didactique de la relation, écologique et sociosémiotique*, Éditions des archives contemporaines. <https://doi.org/10.17184/eac.9782813004055>

MacQueen, J. (1967). « Some methods for classification and analysis of multivariate observations », dans *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, vol. 1, n° 5.1, p. 281-298.

Magnusson, S., Krajcik, J., et Borko, H. (1999). « Nature, sources, and development of pedagogical content knowledge for science teaching », dans *Examining pedagogical content knowledge*, Springer, p. 95-132. [https://doi.org/10.1007/0-306-47217-1\\_4](https://doi.org/10.1007/0-306-47217-1_4)

Margolinas, C. (2014). Connaissance et savoir. Concepts didactiques et perspectives sociologiques ? *Revue française de pédagogie. Recherches en éducation*, vol. 188, p. 13-22. <https://doi.org/10.4000/rfp.4530>

Martinand, J.-L. (1986). *Connaître et transformer la matière*, Peter Lang.

Maturana, H., et Varela, F. (1994). *L'arbre de la connaissance*, Addison Wesley France.

Melin, V. (2019). « Rapport au savoir », dans *Vocabulaire des histoires de vie et de la recherche biographique*, Érès, p. 130-133. <https://doi.org/10.3917/eres.delor.2019.01.0130>

MEN (2016). *Initiation à la programmation, aux cycles 2 et 3*, Éduscol. <https://eduscol.education.fr/document/15409/download>

MEN (2020). *Programmes d'enseignement*, Éduscol. <https://eduscol.education.fr/74/j-enseigne>

Miljanovic, M. A., et Bradbury, J. S. (2018). « A Review of Serious Games for Programming », dans S. Göbel, A. Garcia-Agundez, T. Tregel, M. Ma, J. Baalsrud Hauge, M. Oliveira, T. Marsh, et P. Caserman (dir.), *Serious Games*, Springer International Publishing, vol. 11243, p. 204-216. [https://doi.org/10.1007/978-3-030-02762-9\\_21](https://doi.org/10.1007/978-3-030-02762-9_21)

Mirabail, M. (1990). La culture informatique, *ASTER*, vol. 11, p. 11-28. <https://doi.org/10.4267/2042/9117>

Montiel, H., et Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in k-12 education : A systematic literature review of "scratch" as an innovative programming tool, *Computers*, vol. 10, n° 6, p. 69. <https://doi.org/10.3390/computers10060069>

Montiel, H., et Gomez-Zermeño, M. G. (2022). Rock the Boat! Shaken by the COVID-19 Crisis : A Review on Teachers' Competencies in ICT, *Frontiers in Education*, vol. 6, p. 558. <https://doi.org/10.3389/educ.2021.770442>

Moreno-León, J., Robles, G., et Román-González, M. (2015). Dr. Scratch : análisis automático de proyectos Scratch para evaluar y fomentar el pensamiento computacional, *Revista de Educación a Distancia (RED)*, vol. 46, n° 10. <https://doi.org/10.6018/red/46/10>

Morin, E. (2014). *Introduction à la pensée complexe*, Éditions Points.

Moscovici, S. (2003). « Des représentations collectives aux représentations sociales : éléments pour une histoire », dans *Les représentations sociales*, Presses universitaires de France, vol. 7, p. 79-103. <https://doi.org/10.3917/puf.jodel.2003.01.0079>

Moskal, B. M., Behrens, N., Guzdial, M., Tew, A. E., Dann, W., et Cooper, S. (2006). Computer science assessment instrument development : Evaluating attitudes and outcomes, *STEM Assessment Conference*, p. 194.

Muratet, M. (2023). Comment caractériser et analyse les compétences de la pensée informatique d'un jeu sérieux ?, *EIAH 2023*, p. 178-180. <https://hal.science/hal-04246906/document>

Ndukwe, I. G., et Daniel, B. K. (2020). Teaching analytics, value and tools for teacher data literacy : A systematic and tripartite approach, *International Journal of Educational Technology in Higher Education*, vol. 17, n° 1, p. 22. <https://doi.org/10.1186/s41239-020-00201-6>

- Neumann, K., Nilsson, P., (. . .) Wilson, C. D. (2019). « The refined consensus model of pedagogical content knowledge in science education », dans A. Hume, R. Cooper, et A. Borowski (dir.), *Repositioning Pedagogical Content Knowledge in Teachers' Knowledge for Teaching Science*, Springer, p. 77-94. [https://doi.org/10.1007/978-981-13-5898-2\\_2](https://doi.org/10.1007/978-981-13-5898-2_2)
- Nogry, S., et Spach, M. (2022). Robotics learning at elementary school : Constructing abstractions using multiple instruments, *Frontiers in Education Technology*, vol. 5, n° 2, p. 36. <https://doi.org/10.22158/fet.v5n2p36>
- Nogry, S., Boulc'h, L., et Villemonteix, F. (2019). *Le numérique à l'école primaire : pratiques de classe et supervision pédagogique dans les pays francophones*, Presses universitaires du Septentrion.
- Okoli, C., et Pawlowski, S. D. (2004). The Delphi method as a research tool : an example, design considerations and applications, *Information et Management*, vol. 42, p. 15-29. <https://doi.org/10.1016/j.im.2003.11.002>
- Paillé, P. et Mucchielli, A. (2003). *L'analyse qualitative en sciences humaines et sociales*, Armand Colin.
- Papert, S. (1980). *Mindstorms : Children, Computers, and Powerful Ideas*, Basic Books.
- Papert, S. (1993). *Mindstorms : Children, computers, and powerful ideas* (2nd ed.), Basic Books.
- Parmentier, Y. (2018). Enseigner la pensée informatique à l'école primaire : formation initiale et continue des professeurs, *Atelier « Organisation et suivi des activités d'apprentissage de l'informatique : outils, modèles et expériences » RJC-EIAH*. <https://hal.science/hal-01762626>
- Parriaux, G., Reffay, C., Drot-Delange, B., et Khaneboubi, M. (2023). « Teachers' knowledge in informatics—Exploring educational robotics resources through the lens of textual data analysis », dans J.-P. Pellet et G. Parriaux (dir.), *Informatics in Schools. Beyond Bits and Bytes : Nurturing Informatics Intelligence in Education*, ISSEP 2023, *Lecture Notes in Computer Science*, vol. 14 296, p. 126-138. [https://doi.org/10.1007/978-3-031-44900-0\\_10](https://doi.org/10.1007/978-3-031-44900-0_10)
- Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond : an overview of learner modeling techniques, *User Modeling and User-Adapted Interaction*, vol. 27, n° 3, p. 313-350. <https://doi.org/10.1007/s11257-017-9193-2>
- Pelánek, R. (2018). Exploring the utility of response times and wrong answers for adaptive learning, *Proceedings of the fifth annual ACM conference on learning at scale*, p. 1-4. <https://doi.org/10.1145/3231644.3231675>

Pélisset, É. (1985). Pour une histoire de l'informatique dans l'enseignement français – Premiers jalons, *Système éducatif et révolution informatique*. [Republié dans la *Revue de l'EPI (Enseignement Public et Informatique)*], vol. 50. <https://www.epi.asso.fr/revue/histo/h85ep.htm>

Piaget, J. (1935). *La naissance de l'intelligence chez l'enfant*, Delachaux et Niestlé Neuchatel.

Pólya, G., et Conway, J. H. (1957). *How to solve it : A new aspect of mathematical method*, Princeton University Press.

R Core Team, R. (2022). *R : A language and environment for statistical computing*.

Rabardel, P. (1995). *Les hommes et les technologies, Approche cognitive des instruments contemporains*, Armand Colin.

Reffay, C., Parriaux, G., Drot-Delange, B., et Khaneboubi, M. (2023). « Robotics in primary education : A lexical analysis of teachers' resources across robots », dans T. Keane, C. Lewin, T. Brinda, et R. Bottino (dir.), *Towards a Collaborative Society through Creative Learning*, WCCE 2022, *IFIP Advances in Information and Communication Technology*, vol. 685. [https://doi.org/10.1007/978-3-031-43393-1\\_20](https://doi.org/10.1007/978-3-031-43393-1_20)

Reinert, M. (1983). Une méthode de classification descendante hiérarchique : application à l'analyse lexicale par contexte, *Cahiers de l'analyse des données*, vol. 8, n° 2, p. 187-198.

Reuter, Y. (2004). Analyser la discipline : quelques propositions, *Actes du 9<sup>e</sup> colloque de l'association internationale pour la recherche en didactique du Français*, vol. 35, p. 5-12. <https://doi.org/10.3406/airdf.2004.1611>

Reuter, Y. (2007a). « Forme scolaire », dans Y. Reuter (dir.), *Dictionnaire des concepts fondamentaux des didactiques*, De Boeck, p. 11-15.

Reuter, Y. (2007b). « Discipline scolaire », dans Y. Reuter, C. Cohen-Azria, B. Daunay, I. Delcambre, et D. Lahanier-Reuter, *Dictionnaire des concepts fondamentaux des didactiques*, p. 85-89.

Reuter, Y. (2011). « Penser la perspective didactique : la question de l'articulation entre disciplinaire, pédagogique et scolaire », dans B. Daunay, Y. Reuter et B. Schneuwly (dir.), *Concepts et méthodes en didactique du français*, Presses universitaires de Namur, p. 15-40.

Reuter, Y. (2013). *Un déplacement théorique en didactique(s) : le vécu disciplinaire des élèves, Souffrances, plaisirs et autres émotions des enfants à l'école*, Journée d'étude organisée par le laboratoire EMA, Cergy-Pontoise, France.

Reuter, Y. (2014). Construire la catégorie de discipline scolaire en didactique(s). *Linguarum Arena*, vol. 5, p. 79-95.



- Reuter, Y., et Lahanier-Reuter, D. (2007). L'analyse de la discipline : quelques problèmes pour la recherche en didactique, *9<sup>e</sup> colloque de l'association internationale pour la recherche en didactique du Français*. <https://doi.org/10.3406/airdf.2007.1729>
- Robert, F. (1985). L'utilisation de l'ordinateur dans l'enseignement primaire : l'exemple de la France, *Enfance*, vol. 38, n° 1, p. 19-30. <https://doi.org/10.3406/enfan.1985.2857>
- Robert A. et Robinet J. (1992). Représentations des enseignants et des élèves, *Repères Irem*, vol. 7, p. 93-99.
- Robert, A., et Rogalski, J. (2002). Le système complexe et cohérent des pratiques des enseignants de mathématiques : une double approche, *La revue canadienne de l'enseignement des sciences, des mathématiques et des technologies*, vol. 2, n° 4. <https://doi.org/10.1080/14926150209556538>
- Rogalski, J. (1987). Acquisition de savoirs et de savoir-faire en informatique, *Cahiers de Didactique des Mathématiques*, vol. 43.
- Rogalski, J. (2015). Psychologie de la programmation, didactique de l'informatique. Déjà une histoire. . ., *Informatique en éducation : perspectives curriculaires et didactiques*.
- Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative, *Formation et Profession*, vol. 24, n° 1, p. 87-89. <https://doi.org/10.18162/fp.2016.a92>
- Romero, M., Lepage, A., et Lille, B. (2017). Computational thinking development through creative programming in higher education, *International Journal of Educational Technology in Higher Education*, vol. 14, n° 1, p. 1-15. <https://doi.org/10.1186/s41239-017-0080-z>
- Romero, M., Lille, B., Viéville, T., Duflot-Kremer, M., Smet, C. D., et Belhassein, D. (2018). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché, *Educode - Conférence internationale sur l'enseignement au numérique et par le numérique*. <https://inria.hal.science/hal-01861732>
- Rouquette, M.-L. (1973). *La créativité*, Presses universitaires de France.
- Roy, A. (2015). Les fonctions exécutives chez l'enfant : des considérations développementales et cliniques à la réalité scolaire, *Dev*, vol. 7, p. 13-40.
- Saddoug, H., Rahimian, A., Marne, B., Muratet, M., Sehaba, K., et Jolivet, S. (2022). Review of the Adaptability of a Set of Learning Games Meant for Teaching Computational Thinking or Programming in France, *Proceedings of the 14th International Conference on Computer Supported Education*, p. 562-569. <https://doi.org/10.5220/0011126400003182>



Schrader, P., Mariappan, J., et Shih, A. (2004). Scenario Based Learning Approach In Teaching Statics, *2004 Annual Conference Proceedings*, 9.1083.1-9.1083.7. <https://doi.org/10.18260/1-2-13347>

Séjourné, A., Chen, Y., Voulgre, E., Brilland, X., et Brothier, S. (2023). Formation MEEF LVE-TICE et Humanités numériques, *Productions des groupes thématiques numériques (Direction du numérique pour l'éducation du Ministère de l'Éducation nationale et de la Jeunesse)*. <https://hal.science/hal-03938949v1>

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

Seren Smith, M., Warnes, S., et Vanhoestenbergh, A. (2018). « Scenario-based learning », dans J. P. Davies et N. Pachler (dir.), *Teaching and Learning in Higher Education : Perspectives from UCL*, UCL IOE Press, p. 144-156. <https://www.ucl-ioe-press.com/ioe-content/uploads/2018/05/Teaching-and-Learning-in-Higher-Education.pdf>

Sergis, S., et Sampson, D. G. (2017). « Teaching and Learning Analytics to Support Teacher Inquiry : A Systematic Literature Review », dans A. Peña-Ayala (dir.), *Learning Analytics : Fundamentals, Applications, and Trends*, Springer International Publishing, vol. 94, p. 25-63. [https://doi.org/10.1007/978-3-319-52977-6\\_2](https://doi.org/10.1007/978-3-319-52977-6_2)

Shulman, L. S. (1986). Those who understand : Knowledge growth in teaching, *Educational researcher*, vol. 15, n° 2, p. 4-14. <https://doi.org/10.30827/profesorado.v23i3.11230>

Shulman, L.-S. (2007). Ceux qui comprennent. Le développement de la connaissance dans l'enseignement, *Éducation et didactique*, vol. 1, n° 1, p. 97-114. <https://doi.org/10.4000/educationdidactique.121>

Conseil scientifique de la SIF (2014). L'informatique : la science au cœur du numérique, *Bulletin de la société informatique de France*, vol. 2, p. 13-20. <https://doi.org/10.48556/SIF.1024.2.13>

Simon, J. C. (1980). *L'éducation et l'informatisation de la société : Rapport au président de la République*, Documentation française.

Simonnot, B. (2008). Quand les moteurs de recherche appellent au jeu : usages ou détournements ?, *Questions de communication*, vol. 14, p. 95-114. <https://doi.org/10.4000/questionsdecommunication.752>

Spach, M. (2017). *Activités robotiques à l'école primaire et apprentissage de concepts informatiques : quelle place du scénario pédagogique ? Les limites du co-apprentissage*, thèse de l'université Sorbonne Paris Cité. <https://theses.hal.science/tel-02271924>

Spach, M. (2019). Activités robotiques à l'école : approches de pratiques d'enseignement et effets sur les apprentissages, *Recherches en didactiques*, vol. 28, n° 2, p. 68-87. <https://doi.org/10.3917/rdid.028.0068>

Srikant, R., et Agrawal, R. (1996). « Mining sequential patterns : Generalizations and performance improvements », dans P. Apers, M. Bouzeghoub, et G. Gardarin (dir.), *Advances in Database Technology-EDBT '96*, Springer Berlin Heidelberg, vol. 1057, p. 1-17. <https://doi.org/10.1007/BFb0014140>

Stokes, A., Aurini, J., Rizk, J., Gorbet, R., et McLevey, J. (2023). Using robotics to support the acquisition of STEM and 21st-century competencies : Promising (and practical) directions, *Canadian Journal of Education / Revue canadienne de l'éducation*, vol. 45, n° 4, p. 1141-1170. <https://doi.org/10.53967/cje-rce.5455>

Storey, M.-A., Phillips, B., Maczewski, M., et Wang, M. (2002). Evaluating the usability of web-based learning tools, *Journal of Educational Technology & Society*, vol. 5, n° 3, p. 91-100.

Tardif, J. (2006). *L'évaluation des compétences. Documenter le parcours de développement*, Chenelière Éducation.

Tchounikine, P. (2016). *Initier les élèves à la pensée informatique et à la programmation avec Scratch*. <https://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.pdf>

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M. et Yeo, S. H. (2016). A review on the use of robots in education and young children, *Journal of Educational Technology & Society*, vol. 19, n° 2, p. 148-163.

Touloupaki, S. (2023). Contribution à l'étude de l'apprentissage de la programmation en grande section et en cours préparatoire, à travers le logiciel ScratchJr : une approche didactique exploratoire, thèse de l'université Paris Cité. <https://theses.fr/s177034>

Tremblay, C. et Poellhuber, B. (2022). Analyse qualitative de référentiels de compétences du XXI<sup>e</sup> siècle, numériques et informationnelles : tendances mondiales observées, *Formation et profession*, vol. 30, n° 2, p. 1-26. <https://doi.org/10.18162/fp.2022.648>

UNESCO (2018). *Référentiel UNESCO de compétences TIC pour les enseignants*, UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000368966>

Vahldick, A., Mendes, A. J., et Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies, *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, p. 1-7. <https://doi.org/10.1109/FIE.2014.7044114>

Vandevelde, I., Fluckiger, C., et Nogry, S. (2022). Resources and textbooks for computer science education in French primary

schools, *IARTEM E-Journal*, vol. 14, n° 1, p. 1-20. <https://doi.org/10.21344/iartem.v14i1.954>

Veresov, N. (2014). « Émotions, Perezhivanie et développement culturel : le projet inachevé de Lev vygotski », dans C. Moro et N. Muller Mirza (dir.), *Sémiotique, Culture et Développement Psychologique*, Presses universitaires du Septentrion.

Vergnaud, G. (1989). « La théorie des champs conceptuels », *Publications de l'Institut de recherche mathématiques de Rennes*, n° S6, p. 47-50.

Vergnaud, G. (1991). « La théorie des champs conceptuels », dans *Recherches en didactique des mathématiques*, La Pensée Sauvage.

Vergnaud, G. (2001). Forme opératoire et forme prédicative de la connaissance, *Investigações em Ensino de Ciências*, vol. 17, n° 2, p. 287-304.

Vergnaud, G. (2007). Représentation et activité : deux concepts étroitement associés, *Recherches en éducation*, vol. 4, p. 9-22. <https://doi.org/10.4000/ree.3889>

Vergnaud, G. (2013). Qu'est-ce que la pensée?, *La nouvelle revue de l'adaptation et de la scolarisation*, vol. 63, n° 3, p. 277-299. <https://doi.org/10.3917/nras.063.0277>

Vergnaud, G., et Durand, C. (1976). Structures additives et complexité psychogénétique, *Revue française de pédagogie*, p. 28-43. <https://doi.org/10.3406/rfp.1976.1622>

Villemonteix, F. (2007). *Les animateurs TICE à l'école primaire : spécificités et devenir d'un groupe professionnel : analyse de processus de professionnalisation dans une communauté de pratiques en ligne*, thèse de l'université René Descartes – Paris V. [https://theses.hal.science/file/index/docid/202443/filename/These\\_Francois\\_Villemonteix.pdf](https://theses.hal.science/file/index/docid/202443/filename/These_Francois_Villemonteix.pdf)

Vinck, D. (2009). De l'objet intermédiaire à l'objet frontière. Vers la prise en compte du travail d'équipement, *Revue anthropologique des connaissances*, vol. 3, n° 1, p. 51-72.

Von Uexküll, J. (2010). *Milieu animal et milieu humain*, Bibliothèques Rivages.

Voogt, J., et Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences : Implications for national curriculum policies, *Journal of curriculum studies*, vol. 44, n° 3, p. 299-321. <https://doi.org/10.1080/00220272.2012.668938>

Watson, G. (2006). Technology Professional Development : Long-Term Effects on Teacher Self-Efficacy, *Journal of Technology and Teacher Education*, vol. 14, n° 1, p. 151-166.

- Wilson, A., et Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming, *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest group-PPIG2010*. <https://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf>
- Wilson, A., Hainey, T., et Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts, *Proceedings of the 6th European Conference on Games-Based Learning (ECGBL)*, p. 4-5.
- Wing, J. M. (2006). Computational thinking, *Communications of the ACM*, vol. 49, n° 3, p. 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, vol. 366, n° 1881, p. 3717-3725. <https://doi.org/10.1098/rsta.2008.011>
- Wolz, U., Hallberg, C., et Taylor, B. (2011). Scrape : A tool for visualizing the code of scratch programs, *Poster présenté à la 42<sup>e</sup> ACM Technical Symposium on Computer Science Education (SIGCSE 2011)*, Dallas, États-Unis.



# Table des matières

Préface	v
Introduction	1
<b>PARTIE 1</b>	
<b>ENSEIGNER L'INFORMATIQUE À L'ÉCOLE</b>	<b>11</b>
1 Quel enseignement de l'informatique à l'école primaire en France ? Réflexions sur 40 ans de développements	13
2 Quel enseignement de l'informatique dans la scolarité obligatoire en France ? Analyse des programmes et manuels	27
3 Pratiques envisagées des enseignants pour un enseignement de l'informatique à l'école primaire	51
4 Quelles ressources pour enseigner l'informatique dans le premier degré ? Une étude de cas	69
<b>PARTIE 2</b>	
<b>APPRENDRE L'INFORMATIQUE À L'ÉCOLE</b>	<b>85</b>
5 Ruptures et continuités dans les représentations de l'informatique et de son apprentissage chez les élèves de cycles 3 et 4	87
6 Usages des robots programmables BeeBot en classe : des objets pour apprendre, des objets à apprendre	115
7 Zoom sur quelques erreurs récurrentes lors des premiers apprentissages en algorithmique	135
<b>PARTIE 3</b>	
<b>FORMER LES ENSEIGNANTS</b>	<b>155</b>
8 Enseigner l'informatique à l'école primaire : quelques caractéristiques des représentations des formateurs d'enseignants du premier degré	157

<b>9</b>	<b>Connaissances pour enseigner l'informatique : analyse textuelle de productions d'enseignants de l'école primaire</b>	<b>173</b>
<b>10</b>	<b>Former les enseignants à et par la programmation informatique cocréative en formation initiale</b>	<b>193</b>
<b>11</b>	<b>Assister les enseignants du primaire dans l'enseignement de la pensée informatique. Une approche basée sur la scénarisation et vers les <i>teaching analytics</i></b>	<b>211</b>
	<b>POSTFACE</b>	<b>233</b>
	<b>De modèles d'analyse à un modèle de conception de formations à l'informatique</b>	<b>235</b>
	<b>BIBLIOGRAPHIE</b>	<b>249</b>

Publié par l'université Paris Cité, 85 boulevard Saint-Germain 75006 Paris

Édition : Alice Trotignon, Vincent Colpin  
(Direction Générale déléguée aux Bibliothèques et Musées, Université Paris Cité)

Impression : Ateliers de l'université Paris Cité

Dépôt légal : septembre 2024

Imprimé sur papier FSC





# Enseigner, apprendre, former à l'informatique à l'école : regards croisés

Édition coordonnée par Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay

Cet ouvrage présente un ensemble de recherches sur les conditions d'enseignement-apprentissage de l'informatique à l'école primaire. Mobilisant des chercheurs en éducation, informatique, psychologie et linguistique, il examine les défis spécifiques de l'enseignement de l'informatique à l'école à travers une étude des politiques publiques, des pratiques enseignantes, et des perceptions des élèves et des enseignants. Il propose également des ressources pédagogiques et des outils de scénarisation pour soutenir l'enseignement de la pensée informatique, visant à combiner rigueur scientifique et diffusion accessible des savoirs.

L'ouvrage *Enseigner, apprendre, former à l'informatique à l'école : regards croisés* s'adresse à un large public, y compris les chercheurs, les formateurs et les enseignants intéressés par l'enseignement de l'informatique.

