



HAL
open science

Enseigner l'informatique à l'École : à la recherche d'une raison d'être

Vanea Chiprianov, Lalina Coulange, Grégory Train²

► To cite this version:

Vanea Chiprianov, Lalina Coulange, Grégory Train². Enseigner l'informatique à l'École : à la recherche d'une raison d'être. 1024 : Bulletin de la Société Informatique de France, 2018. hal-02514250

HAL Id: hal-02514250

<https://hal.science/hal-02514250>

Submitted on 21 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Enseigner l'informatique à l'École : à la recherche d'une raison d'être...

Vanea Chiprianov¹, Lalina Coulange² et Grégory Train²

Introduction

L'entrée de l'enseignement de l'informatique à l'école (primaire et collège) s'est accompagnée d'un intérêt croissant bien au-delà des frontières de l'école. Ainsi, la sphère médiatique se fait ici et là l'écho d'illustres avis sur le bien-fondé d'un tel apprentissage, ces avis s'avérant parfois contradictoires : pour les uns, le code informatique est « plus important que d'apprendre l'anglais » (Tim Cook, cité dans *Le Monde*), pour les autres, à l'instar de Laurent Alexandre, « les plans de formation massifs de développeurs informatiques conduiront à beaucoup de désillusions car le code informatique bas de gamme sera entièrement automatisé » (Laurent Alexandre, *La guerre des intelligences*, Éditions JC Lattès). En tentant une prise de distance vis-à-vis de ces positions parfois tranchées, dénotant peut-être plus encore de la question délicate des relations à construire demain entre l'homme et la machine, cet article vise à poser les jalons d'une problématique de recherche didactique, liée à l'enseignement et à l'apprentissage de l'informatique à l'École.

Nous commençons par retracer des étapes dans l'histoire conjointe de l'enseignement et des recherches en didactique de l'informatique de 1976 à aujourd'hui. Nous cherchons ensuite à formuler un ensemble de questions relatives au contexte d'étude contemporain de la science informatique. Ces questions nous conduisent à

1. Laboratoire d'informatique de l'université de Pau et des Pays de l'Adour.

2. Laboratoire Lab-E3D (EA 7441), université de Bordeaux.

rechercher une « raison d'être » (Chevallard et Wozniak, 2003) des savoirs informatiques à enseigner, que nous considérons comme une garantie didactique essentielle de préservation de l'authenticité de ces savoirs.

Précisons qu'il existe de nombreux regards historiques posés sur l'enseignement de l'informatique à l'école (Archambault 2013, 2017 ; Baron et al., 2014 ; Baron et Drot-Delange 2016). Parce qu'il serait vain d'enrichir de tels travaux, notre démarche consiste en l'adoption d'une focale particulière³. Le point de vue historique retenu se situe au croisement de l'histoire de la transposition didactique des savoirs informatiques à enseigner (et de l'étude des conditions qui pèsent sur cette transposition didactique) et de l'histoire des recherches en didactique de l'informatique. L'enjeu n'est pas de chercher une forme d'exhaustivité dans cet historique mais de repérer des dynamiques dans la transposition didactique des savoirs à enseigner en informatique, en lien avec l'histoire de cette discipline scolaire et des recherches conduites sur sa scolarisation.

Précisons également que notre démarche s'inscrit dans une perspective d'étude de la transposition didactique des savoirs informatiques. Si le point de vue historique permet dans un premier temps de mieux comprendre les conditions et contraintes spécifiques (à la fois celles qui ont pu exister et celles en œuvre aujourd'hui) liées à cette transposition, il n'épuise pas la question. En particulier, la recherche de raisons d'être des savoirs informatiques constitue un problème premier de transposition didactique posé à la profession enseignante : comment développer un enseignement scolaire de l'informatique qui apparaisse à la fois fidèle à l'informatique (en tant que science) et pertinent pour la formation scolaire des élèves (en tant que futurs citoyens). C'est donc cette voie que nous empruntons également, éclairée par ce premier point de vue historique.

Un regard historique croisé

Dans cette section, nous évoquons, de manière conjointe, le contexte de l'informatique comme discipline scolaire et celui des recherches en didactique associées. Ce travail est conduit en prise d'appui des travaux menés par Rogalski (2015) dans le domaine de la psychologie de la programmation⁴, précurseur de la didactique de l'informatique (de la Higuera, 2016), en proposant de les compléter. Notre perspective est l'identification de dynamiques à même de fournir un premier éclairage sur les conditions et les contraintes susceptibles de peser sur la mise en œuvre de la décision récente (MENSUR, 2015) de (ré)introduire massivement l'enseignement de l'informatique dans les programmes scolaires.

3. Nécessairement partielle mais en veillant à éviter la partialité.

4. Dénomination retenue par l'auteur.

L'histoire de l'enseignement de l'informatique dans les écoles, collèges et lycées et l'histoire des recherches en didactique de l'informatique (Rogalski, 2015), (Nijimbere, 2015, pp. 125-145) sont intrinsèquement liées, l'enseignement constituant l'objet principal de ces recherches. Rogalski (2015) identifie quatre générations de recherche, chacune s'intéressant à des thématiques spécifiques.

Première génération (1967 – 1972)

L'enseignement de l'informatique en France, contrairement à d'autres pays (par exemple les États-Unis où il est issu de l'enseignement de l'électronique), trouve sa source majoritairement dans l'enseignement supérieur (en mathématiques) dans les années 1960 (Nijimbere, 2015). Une première vague de travaux se situe dans le domaine de la psychologie ergonomique et procède d'une approche avant tout prescriptive. Il s'agit d'étudier des concepts clés de programmation à travers des questionnements liés à la résolution de problèmes et à l'organisation de l'activité engagée dans des tâches de compréhension/production de programmes et de planification. Ces travaux sont caractérisés par des problématiques liées à la construction d'expertises notamment à travers des comparaisons novices/experts en termes de connaissances, d'organisation de l'action et d'analyse des erreurs. Des problématiques liées aux interactions homme-machine (représentations – modèles mentaux – du dispositif informatique, utilisation des aides à la programmation, etc.) et aux apprentissages sous-jacents sont également abordées.

Deuxième génération (années 1970)

Ces travaux sont principalement des études expérimentales conduites par des informaticiens (*computer scientists*). En France, cette décennie marque l'introduction de l'enseignement de l'informatique au lycée (Baron et al., 2014), avec l'expérience des 58 lycées : 58 lycées sont équipés de mini-ordinateurs et un total d'environ 7000 enseignants (Nijimbere, 2015), d'horizons différents, bénéficient d'un plan de formation sous la responsabilité de spécialistes universitaires d'informatique. En l'absence de programme scolaire spécifique, l'accent est mis sur la construction d'un mode de penser spécifique à l'informatique et sur le triptyque : organisation, modélisation, algorithmique. Cette période se caractérise par la non distinction de l'enseignement de l'informatique en tant que science en soi (ce que nous appelons science informatique – SI) et en tant qu'outil au service des autres disciplines (ce que nous appelons, en empruntant le terme au rapport de l'Académie des sciences (AdS, 2013), alphabétisation numérique – AN⁵). Cette dialectique SI – AN sera dès lors d'importance dans la construction de l'enseignement de l'informatique en France.

5. Ce concept est proche de celui de « littératie numérique », toutes les deux proches du terme anglais *digital literacy*.

Troisième génération – l'âge d'or (années 1980)

Suite à cette première expérience, l'enseignement de l'informatique continue à se développer en France (Baron et al., 2014). En 1981, différents rapports mettent l'accent sur les différences entre science informatique (SI) et alphabétisation numérique (AN) (Simon, 1980). Le ministère décide d'introduire au lycée une option informatique (SI) sous la responsabilité d'enseignants par ailleurs non spécifiquement formés. Les instructions officielles qui accompagnent la mise en place de l'option mettent l'accent sur le mode de pensée algorithmique. Algorithmique et programmation constituent le noyau central du programme, dans une démarche de transposition assez directe des principales parties du programme de la licence informatique. Le succès est tel que de nombreux élèves se voient refuser l'accès à cette option, la demande dépassant largement l'offre. Néanmoins, cette option, « présente dans un lycée sur deux et en voie de généralisation, [...] est supprimée en 1992, rétablie en 1995 puis à nouveau supprimée en 1998 » (Archambault, 2013). « Les conditions de création d'une nouvelle discipline scolaire informatique semblaient alors réunies, mais la volonté politique n'a pas été au rendez-vous (*Creating a new discipline might well have been possible had there been sufficient political will*) » (Baron et al., 2014).

Durant cette période, les recherches en didactique de l'informatique sont prolifiques. Elles sont publiées par exemple dans les actes⁶ de cinq colloques francophones sur la didactique de l'informatique, organisés par l'Association francophone pour la didactique de l'informatique (AFDI), 1988–1996, ce qui a donné lieu à un effort de synthèse (Baudé, 2016). Ces recherches investissent des directions multiples, encore d'actualité pour certaines (représentation des dispositifs, expression des procédures de traitement-planification, schémas de programmes, programmation d'automates, structures conditionnelles, écriture et interprétation de procédures récursives...). Les liens entre programmation et apprentissages scolaires sont également l'objet de nombreux travaux centrés sur des analyses conceptuelles et/ou épistémologiques. Ils concernent à la fois les acquisitions cognitives en lien avec l'apprentissage de la programmation et des systèmes de commande. En particulier, les questions soulevées par la représentation et le traitement des variables (déclaration, affectation, contrôle dans les instructions de choix et itératives) sont largement documentées.

Processus de tarissement (années 1980)

Une rupture s'opère en France dès 1986. Alors que le plan informatique pour tous (IPT) permet en 1985 l'équipement d'écoles, de collèges et de lycées avec 120 000 micro-ordinateurs, et la formation d'environ 110 000 enseignants à l'usage des matériels, logiciels et aux activités de programmation dans des stages de 50 heures

6. <https://www.epi.asso.fr/association/dossiers/epi-afdi.htm>, liens consultés en mai 2018.

(Archambault, 2015), il est jugé sévèrement, au regard notamment des choix matériels (micro-ordinateurs peu adéquats dans l'exécution des logiciels pédagogiques) et au regard d'une formation évaluée comme insuffisante des enseignants, avec des conséquences parfois fâcheuses : « les micro-ordinateurs finissant rapidement leur carrière au placard » (Brulé, 1993).

Le souhait d'intégrer l'enseignement de l'informatique dans l'enseignement d'autres disciplines, sensiblement moins coûteuse que la création d'une nouvelle discipline, s'affirme. L'idée d'apprendre « en faisant » (*A deceptive idea [...] "learning by doing"*) (Baron et al., 2014), dans une économie de prise en charge de conceptualisation, conduit, en 1992, à la substitution de l'enseignement optionnel d'informatique par des ateliers pratiques d'alphabétisation numérique (AN). L'informatique y est appréhendée exclusivement dans son versant outil, mis à disposition pour l'enseignement des autres disciplines.

L'enseignement de l'informatique disparaît complètement du curriculum officiel en 1999. Ceci s'accompagne d'un tarissement des recherches dans les domaines de la psychologie de la programmation et de la didactique de l'informatique. Les causes (Rogalski, 2015) de ce tarissement tiennent à la fois à l'arrêt des enseignements qui va de pair avec la disparition de terrains d'étude. Elles tiennent peut-être également aux tentatives infructueuses d'impliquer doublement les chercheurs en didactique des mathématiques ou des sciences et les chercheurs en informatique, avec des frontières disciplinaires difficiles à dépasser.

Quatrième génération – Computer Science Education (années 2000)

Au début des années 2000, un nouveau revirement prend forme dans l'enseignement de l'informatique. En 2001, une certification des compétences d'utilisation des outils informatiques au collège et au lycée – le Brevet informatique et internet (B2i) – est créée. Quand bien même ce certificat reste centré sur l'alphabétisation numérique, il dénote un premier regain d'intérêt de l'informatique à l'école (Baron et al., 2014). En 2004, dans une forme de continuité, la Certification informatique et internet (C2i) vient compléter le B2i dans l'enseignement supérieur, au niveau licence. L'obtention de ce certificat est d'ailleurs, pendant un temps, exigée institutionnellement pour entamer la carrière d'enseignant de toute discipline. Si la création successive de ces deux certificats montre ainsi une prise de conscience croissante de l'importance de l'informatique, ils demeurent cependant fortement tournés sur l'AN d'une manière pratique, en délaissant l'aspect SI.

Les travaux en didactique de l'informatique durant cette période sont pourtant caractérisés par un renouveau des recherches sur la programmation ainsi que sur les concepts propres de la SI. Succédant aux colloques organisés par l'AFDI, les colloques francophones Didapro-DidaSTIC⁷, qui investissent initialement des questions didactiques liées aux logiciels, intègrent progressivement

7. <http://didapro.org/>

les Sciences et technologies de l'information et de la communication (STIC) et la SI à cette réflexion. Ces recherches sont par ailleurs principalement orientées vers l'enseignement de l'informatique universitaire et produits essentiellement par les informaticiens (*computer scientists*) et par quelques chercheurs en psychologie cognitive (*cognitive science*) (Rogalski, 2015). Les thématiques de recherche s'articulent autour de questions d'accessibilité des concepts de programmation, de difficultés en matière de modèles mentaux, des différents rôles joués par les variables informatiques ou encore des difficultés rencontrées pour déboguer un programme. On trouve également une reprise de travaux plus anciens sur la récursivité dans un cadre piagétien étendu à la conceptualisation de haut niveau. Des questions spécifiques relatives aux nouvelles possibilités de programmer voient le jour. Elles s'accompagnent d'un déplacement de la problématique des langages en tant que tels, aux paradigmes de programmation et aux rôles que ces paradigmes sont susceptibles d'endosser. Il est ainsi question du paradigme de la programmation orientée objet et des problèmes d'abstraction et de représentation des données qu'il pose – il est aussi question du paradigme de la programmation en parallèle avec des questions partiellement renouvelées de représentation du dispositif informatique. De manière générale, les concepts propres à la SI redeviennent les entrées dominantes dans les recherches de cette décennie, ce qui peut paraître paradoxal au regard de ce qui se produit dans le même temps dans l'institution scolaire.

Vers une cinquième génération ? (années 2010)

Au niveau international, à la fin des années 2000, un mouvement (Wing, 2006) de valorisation de l'enseignement de l'informatique prend de l'ampleur. Ce mouvement gravite autour du concept de la « pensée informatique » (*computational thinking*). Alors que la recherche d'un consensus sur la définition de ce concept est toujours d'actualité, celui-ci semble graviter autour des savoirs et savoir-faire attachés à la SI : résolution de problèmes, conception de systèmes, logique conditionnelle, pensée itérative, parallèle et récursive, etc. Mais il apparaît également que ce concept de pensée informatique s'inscrit dans une vision plus large, non réduite au seul contexte informatique, avec des applications dans d'autres contextes et disciplines. L'argumentaire développé semble ériger la pensée informatique en un savoir-faire fondamental, nécessaire à tout individu, et non plus réservé aux seuls informaticiens. Cette dynamique va de pair avec une focale s'orientant vers la SI. De nombreux pays, notamment au niveau des décideurs politiques, apparaissent sensibilisés à un tel argumentaire : la Grande-Bretagne, les États-Unis, Israël, la Nouvelle-Zélande (Brown et al., 2014), (Gal-Ezer and Stephenson, 2014), (Bell et al., 2014) et plus récemment, la France...

En France, en 2012, suite à une action concertée de l'EPI (association Enseignement public et informatique), de la SIF (Société informatique de France) et du groupe ITIC-EPI-SIF (Archambault, 2013), un cours optionnel d'Informatique et sciences

du numérique (ISN) est introduit au lycée (Baron et al., 2014). L'algorithmique et la programmation sont au programme de cette option. Un signal essentiel est donné avec le rapport de l'Académie des sciences (AdS, 2013). Ce rapport, en évoquant des raisons principalement sociétales et en mettant en exergue le retard pris en la matière par rapport à d'autres pays, affirme la nécessité d'introduire l'enseignement de la SI dans les enseignements primaires et secondaires. Une esquisse de curriculum ainsi que des recommandations relatives à la nécessité d'une formation des enseignants sont proposées. En mai 2015, le président français annonce⁸ un budget d'un milliard d'euros dédié à l'initiation des élèves, de l'école primaire jusqu'au lycée et à la formation initiale et continue des enseignants. Ce sont des dimensions, concernant à la fois l'AN et la SI, qui sont déclinées dans un programme ambitieux en place depuis la rentrée 2016 dans le cadre de la scolarité obligatoire, et dont la mise en place se joue actuellement au lycée.

Du côté de la recherche en didactique de l'informatique, cette nouvelle période semble laisser poindre une forme de foisonnement. Des recherches et recherches-actions nouvelles s'initient en prise d'appui sur des initiatives nationales⁹ de diffusion du numérique dans l'éducation. Des associations, dans une certaine mesure nouvelles, voient le jour. Il en va ainsi de la création en novembre 2014 du groupe de réflexion de la CFEM (Commission française pour l'enseignement des mathématiques) sur les relations entre l'enseignement des mathématiques et de l'informatique, regroupant, aux côtés de mathématiciens et didacticiens des mathématiques, des experts de la SIF (Société informatique de France) et de l'EPI (Enseignement public et informatique). Il en va également de la création récente de la C3I (Commission inter-IREM informatique) visant à fédérer les travaux du réseau des IREM (Instituts de recherche sur l'enseignement des mathématiques) sur les questions liées à l'informatique comme discipline et regroupant une pluralité d'acteurs. Cette montée en puissance des recherches, dont la focale s'oriente sensiblement vers la SI, s'accompagne de la mise à disposition de terrains d'expérimentation plus denses (de l'école à l'université) et de matériels informatiques renouvelés dans une certaine mesure (robots éducatifs, logiciels éducatifs de programmation, etc.).

Discussion

Ce rapide tour d'horizon de l'enseignement de l'informatique à l'école et des recherches associées laisse se dessiner quelques dynamiques particulièrement saillantes de cette histoire commune. En particulier, concernant la dialectique SI/AN, une mise en tension apparaît à la fois dans les savoirs à enseigner et dans les recherches. En parallèle de cette dynamique en tension SI/AN, se dégagent

8. <https://www.gouvernement.fr/action/l-ecole-numerique>

9. Par exemple l'appel à projets e-FRAN (Espaces de formation, recherche et animation numérique) du programme d'investissements d'avenir (PIA) constitue un investissement scientifique de grande ampleur dans la mise en œuvre du plan pour le numérique à l'école.

des trajectoires initiées et avortées d'autonomisation de l'informatique en tant que discipline scolaire, et la question récurrente du vivier d'enseignants et de leur formation.

Dans notre perspective d'étude de la transposition didactique des savoirs informatiques, ces premières dynamiques s'interprètent comme autant de conditions spécifiques liées à cette transposition et des postures distinctes : l'intégration de ces savoirs dans des compétences « transversales numériques » (B2I, etc.) ; leur intégration comme domaines d'étude dans des disciplines scolaires existantes (mathématiques, technologie...) ou encore, dans une perspective plus marquée de prise d'autonomie de la discipline informatique, leur intégration dans des enseignements optionnels (option informatique, plus récemment ISN). En tout état de cause, et même si elle apparaît plus ou moins marquée, la problématique des enseignants chargés de ces enseignements demeure commune.

Dans le même temps, ce que donne à voir ce tour d'horizon historique est un paysage contemporain conjuguant des éléments nouveaux qui sont autant de conditions spécifiques et dans une certaine mesure inédites de transposition : une volonté politique forte et partagée, un nouvel équilibre AN/SI, des associations d'enseignants et de chercheurs d'horizons différents aux intérêts conjoints et convergents, de nouveaux terrains d'étude étendus et une prise en compte de la nécessité de formation des enseignants.

L'histoire est une source inépuisable en ce qu'elle permet de réinterroger et d'interpréter le passé à l'aune de nouvelles connaissances mais aussi en ce qu'elle permet d'éviter les clichés. En cela, il serait bien hasardeux à ce stade de qualifier de « favorables » les conditions qu'offre la situation actuelle, d'une part, parce que l'histoire nous dit aussi que des conditions proches (et précisément qualifiées alors de favorables) ont existé sans pour autant aboutir à une autonomisation de la discipline informatique, et d'autre part, parce que le caractère inédit¹⁰ des conditions actuelles n'est pas une garantie définitive d'un changement.

C'est dans cette perspective prudente que nous proposons dans un premier temps, à partir de ce que l'histoire de l'informatique dans l'enseignement donne à voir comme mouvements, continuités et ruptures, d'éclairer et de tenter de mieux comprendre les enjeux de l'enseignement de l'informatique à l'école aujourd'hui. Dans un second temps, il s'agira de préciser ce qui nous apparaît être des conditions spécifiques manquantes d'une transposition didactique des savoirs informatiques et en particulier, celles liées à la question des *raisons d'être* de l'enseignement de l'informatique.

10. Chaque époque a, de ce point de vue, son lot d'inédit.

L'enseignement de l'informatique à l'école aujourd'hui en question

Des premières questions

Dès lors que l'on s'intéresse à la problématique de l'arrivée actuelle de l'enseignement de l'informatique, un premier lot de questions se pose : pourquoi introduire cet enseignement dans les programmes scolaires de l'école primaire et du collège ? Pourquoi avoir retenu l'algorithmique et la programmation parmi des domaines savants *possibles* de la discipline informatique ? Pourquoi faire advenir ce changement à la rentrée 2016 ? Pourquoi avoir positionné l'enseignement de l'informatique principalement (MENSUR, 2015) dans des programmes de cycle 2, 3 et 4 de mathématiques¹¹ ?

Pourquoi introduire l'enseignement de l'informatique dans les programmes scolaires ?

En France, l'enseignement de l'informatique a su trouver une place à différents moments et dans des formes diverses (à la fois dans les contenus enseignés et dans des configurations différentes) : la SI (algorithmique et programmation) dans les années 70 et surtout 80, l'AN surtout dans les années 2000, mais déjà en germe dans les années 90. L'introduction de l'informatique à l'école (cycles 2, 3 et 4) en 2016 relève ainsi d'une nouvelle tentative, un nouvel épisode d'une série plus longue, débutée dans les années 1970. De ce point de vue, la démarche initiée aujourd'hui s'inscrit dans une forme de continuité.

Pourquoi avoir retenu l'algorithmique et la programmation parmi des domaines savants « possibles » de la discipline de l'informatique ?

Du point de vue des choix opérés, une forte proximité avec le cas de l'option informatique introduite au lycée dans les années 80 s'affiche. Le programme de cette option était centré sur l'algorithmique et la programmation, dans une démarche de transposition assez directe de ce qui était considéré comme l'essentiel du programme de licence universitaire informatique. L'entrée dans les programmes en 2016 entretient de ce point de vue de grandes similarités, avec un curriculum pensé comme résultant d'une transposition du même type.

Pourquoi faire advenir ce changement à la rentrée 2016 ?

Le mouvement initié autour de la pensée informatique en 2006 a progressivement pris de l'ampleur au niveau international, surtout dans les années 2013–2014. Les

11. Au cycle 4, l'enseignement de l'informatique est positionné aussi dans l'enseignement de la technologie (avec des savoirs autour de, par exemple, l'organisation des systèmes de calcul ou encore les réseaux), mais nous nous intéressons ici surtout au positionnement dans les parties des programmes dédiées aux mathématiques, aux cycles 2, 3 et 4, largement plus important que dans d'autres parties et offrant une continuité inter-cycles.

décideurs français semblent s'être inscrits dans ce mouvement, avec une annonce en 2015 et une mise en pratique dès la rentrée suivante.

Pourquoi avoir positionné cet enseignement principalement dans des programmes de mathématiques ?

Cette question nécessite une attention plus particulière. Si la décision d'introduire l'enseignement de l'informatique dans les programmes de mathématiques est liée probablement au contenu des enseignements dispensés – algorithmique et programmation – ces premiers arguments ne résistent que faiblement à l'objection de confier alors ces mêmes enseignements aux professeurs spécialisés dans la SI. C'est un problème de disponibilité de cohorte de potentiels enseignants que révèle l'histoire de l'enseignement de l'informatique. Depuis les années 90, les formations à destination des enseignants dans le champ de la SI ont fait place à des formations centrées sur l'AN, notamment dans le cadre du C2i. La situation à la rentrée 2016 est celle d'une quasi absence d'enseignants spécialisés en informatique.

Sur ce point, le rapport de l'Académie des sciences (AdS, 2013, 7, 28) préconise et même recommande, « dans une perspective d'affirmation de l'autonomie de l'informatique en tant que nouvelle discipline scolaire » (Artigue et Raoult, 2013), « la création des concours de recrutement et des diplômes identiques à ceux des autres sciences au collège et au lycée » – CAPES (éventuellement avec une discipline majeure et une mineure) et Agrégation. Ce même rapport propose également, pour les professeurs des écoles, l'inclusion de la SI dans la formation initiale et dans la formation continue des professeurs en activité dans le cadre d'un développement professionnel qui reste volontariste. En attendant la formation d'une masse critique – comme celle atteinte au milieu des années 1980 – d'enseignants spécialisés, « un recours à des professeurs déjà formés dans des disciplines préexistantes est inéluctable » (Artigue and Raoult, 2013). Les enseignants de mathématiques ont été les plus volontaires pour prendre en charge l'enseignement ISN créé en 2012. En outre, comme le confirme d'autres analyses, notamment l'argumentaire présent dans les programmes scolaires (Haspekian, 2012) ou encore celui développé dans le rapport de l'Académie des sciences (Ads, 2013), le lien historique et épistémologique entre l'algorithmique et les mathématiques, ou l'omniprésence des algorithmes en mathématiques, justifierait l'enseignement de l'informatique dans le cadre des mathématiques.

Pour l'heure, il semble que la solution retenue a été d'ouvrir une option informatique au sein du concours de recrutement des enseignants de mathématiques (CAPES). Si cette solution intermédiaire peut permettre de disposer rapidement¹² d'une cohorte d'enseignants spécialisés, elle est, du point de vue de l'histoire mouvementée de l'enseignement de l'informatique à l'école, dans une certaine mesure teintée de prudence, en renonçant à la constitution d'un corps professoral nouveau

12. mais cela reste à confirmer.

(d'enseignants d'informatique) dans un contexte économique morose, mais au détriment peut-être de l'affirmation d'autonomie de l'informatique en tant que discipline scolaire. Demeure la question de la programmation... Devrait-elle être cantonnée exclusivement à l'enseignement de la technologie, comme semble le suggérer le découpage proposé dans les programmes du cycle 4 (MENSUR, 2015) ?

Ces premières considérations conduisent inéluctablement à interroger les possibles conséquences d'une introduction de l'enseignement de l'informatique dans les programmes de mathématiques de cycle 2, 3 et 4 sur l'enseignement de l'informatique lui-même...

Quelles possibles conséquences d'une introduction de l'enseignement de l'informatique dans les programmes de mathématiques des cycles 2, 3 et 4 ? Et quels besoins ?

L'expérience récente de l'introduction de l'enseignement ISN¹³ au lycée est premièrement informative de ce point de vue. Haspekian (2012) souligne que cette introduction ne s'est pas faite sans difficulté : surcroît de travail des enseignants, manque d'expertise et de formation dans le domaine de l'algorithmique, difficultés techniques et matérielles, tensions entre l'algorithmique-informatique et l'algorithmique-mathématique... Nous retenons en particulier l'existence de tensions entre l'algorithmique-informatique et l'algorithmique-mathématique qui demeure pleinement d'actualité : si l'algorithmique-mathématique peut justifier l'enseignement de l'informatique dans le cadre des programmes de mathématiques, ce sont les spécificités de l'autre, et ses liens étroits avec la programmation, qui sont susceptibles d'être problématiques pour des enseignants non spécialistes de la SI. Ce fait justifie à lui seul ce que Artigue et Raoult (2013) qualifient de « bousculement » dans l'enseignement des mathématiques. Dès lors, les perspectives proches de l'enseignement de l'informatique à l'école semblent possiblement ternes, avec un risque de centration des enseignements sur l'algorithmique-mathématique et une relégation au second plan d'un enseignement de la programmation.

Du côté de la recherche, les besoins se dessinent comme importants. Suite au phénomène de *tarissement* à l'œuvre dans les années 90, la communauté de chercheurs en didactique de l'informatique semble s'être majoritairement réorientée, laissant la place libre aux informaticiens universitaires dans les années 2000. L'introduction de la SI dans les programmes scolaires ouvre la voie à un retour des recherches en didactique de l'informatique (DDI), partiellement renouvelées dans un nouveau contexte scolaire.

On peut s'attendre à ce que la communauté des chercheurs puisse se repeupler, en prise d'appui sur une ancienne garde et son expertise déjà constituée dans les années

13. Même si le programme de l'ISN est « construit autour de quatre parties : représentation de l'information, algorithmique, langages et programmation, architectures matérielles », dans le contexte de l'enseignement de l'algorithmique et de la programmation dans les programmes de cycle 2, 3 et 4 des mathématiques, nous nous concentrons sur l'analyse de ces domaines.

1980, mais encore avec de nouveaux chercheurs, d'horizons divers, à la fois en didactique des mathématiques (DDM) et en didactique de l'informatique (DDI) et de leur mise en synergie. Pour aider la reconstruction de cette communauté, agir sur les causes qui avaient été identifiées comme étant à l'origine du phénomène de tarissement observé dans les années 90, en particulier celles liées peut-être au manque de valorisation des recherches situées dans un entre-deux peu considéré, ni en (didactique des) mathématiques, ni en informatique, apparaît comme une nécessité. En tout état de cause, cette reconstruction est une occasion donnée de reprendre, réinvestir et revaloriser les travaux de recherche en DDI des années 1980.

Discussion

L'introduction de l'enseignement de l'informatique dans les programmes scolaires à la rentrée 2016 s'inscrit dans une forme de continuité de l'histoire de l'enseignement de l'informatique en France. Elle se caractérise par quelques spécificités liées à l'introduction de cet enseignement dans le programme des mathématiques du cycle 4, et un choix d'enseigner des éléments de la SI, en premier lieu l'algorithmique et la programmation.

Si ce premier questionnement donne un éclairage sur les raisons qui ont conduit à la (ré)introduction de l'enseignement de l'informatique dans les programmes scolaires et les conditions et les contraintes susceptibles de peser sur une telle mise en œuvre, reste qu'il est une condition première à la transposition de savoirs disciplinaires, une condition essentielle à l'autonomisation d'une discipline en tant que discipline scolaire : la question de la nécessité d'enseigner cette discipline. Chevallard et Wozniak (2003) érigent cette nécessité en la recherche de raison d'être des savoirs enseignés. Dans le cadre des mathématiques¹⁴, ces auteurs montrent par exemple, dans le cas particulier du domaine de la statistique, en quoi le déficit d'un tel questionnement tend à aplanir l'enseignement à la visite de savoirs sans en identifier le but... Nous pensons, à l'instar de ces auteurs, que la recherche de raisons d'être des savoirs informatiques constitue un problème premier de transposition didactique posé à la profession : comment développer un enseignement scolaire de l'informatique (ou de domaines particuliers de l'informatique) qui apparaisse à la fois fidèle à l'informatique et pertinent pour la formation scolaire des élèves (en tant que futurs citoyens). Ce problème nous semble d'autant plus saillant que d'une part la profession enseignante (autrement dit, les enseignants susceptibles d'enseigner l'informatique) est particulièrement hétérogène et d'autre part, les premiers phénomènes de transposition à l'œuvre dans les programmes sont dans une démarche de

14. Cadre dans lequel la question des raisons d'être d'un savoir est trop rapidement réglée en avançant un argument d'importance de la chose... Nous invitons par exemple le lecteur à rechercher les raisons d'être de l'enseignement du théorème de Pythagore sans emprunter le chemin trop court du « parce que c'est important ».

transposition assez directe de ce qui est considéré comme l'essentiel du programme de licence universitaire informatique et dans une forme de mimétisme du passé (avec l'introduction de l'option informatique dans les années 80).

Dans la section suivante, nous cherchons donc à mieux investir ce qui pourrait constituer des raisons d'être de l'introduction de l'enseignement de l'informatique et en particulier de la science informatique à l'école.

À la recherche des motivations de l'enseignement de l'informatique

Un premier niveau de justifications de l'enseignement de l'informatique

Parmi les justifications avancées plaçant en faveur de l'introduction de la science informatique à l'école, on trouve en premier lieu l'article d'influence de Wing (2006). La pensée informatique y est présentée d'emblée comme une connaissance, un savoir-faire fondamental pour tout un chacun, pas seulement pour les informaticiens, au même rang que les savoir-écrire, lire et compter. De multiples exemples dans lesquels interviennent peu ou prou les concepts informatiques viennent étayer cet argumentaire. La perte d'un objet et le fait de retracer nos pas pour le retrouver est ainsi par exemple apparenté à du *backtracking*. L'influence de l'informatique dans la manière d'appréhender et résoudre des problèmes est également mise en avant : dans des disciplines toutes aussi diverses que la biologie, l'économie, la chimie ou encore la physique, l'usage de l'informatique selon (Wing, 2006, p. 34) modifie en profondeur les représentations qu'ont les scientifiques et les professionnels de ces mêmes disciplines, vers des représentations plus informatisées qui utilisent des concepts informatiques. Compte tenu de l'omniprésence de l'informatique dans tous les domaines de la société, elle estime que, dans le futur proche, ce phénomène sera généralisé : les représentations de tout un chacun, des outils et des objets qu'on utilise dans sa vie professionnelle et privée, comprendront des éléments informatiques, qui iront au-delà de la simple utilisation des ordinateurs (« *Computational biology is changing the way biologists think. Similarly, computational game theory is changing the way economists think; nanocomputing, the way chemists think; and quantum computing, the way physicists think. This kind of thinking will be part of the skill set of not only other scientists but of everyone else.* ») (Wing, 2006, p. 34). Autrement dit, « la question se pose bien de savoir quelles sont les représentations mentales opérationnelles, les connaissances scientifiques et techniques informatiques qui permettent à tout un chacun d'être en phase et en prise sur la société dans laquelle il vit » (Archambault, 2017).

La pensée informatique est un concept avec une longue histoire, présentée par exemple dans Tedre et Denning (2016). Ainsi, une recherche identitaire au début de l'histoire de la discipline (décennies 1950 et 1960) a également intégré une exploration de ce qui était unique dans la manière de « penser informatique ». Des nombreux

auteurs, de Perlis, en passant par Forsythe, Knuth et Dijkstra, ont souligné l'idée que l'identité disciplinaire de l'informatique survient de ses processus mentaux uniques, que la valeur de la programmation est surtout en tant qu'outil pour comprendre tout type de problèmes, pour cultiver un certain style de penser et de concevoir des solutions. Des arguments en faveur d'une pensée algorithmique entraînant le cerveau à être un meilleur solveur de problèmes dans tous les domaines, en fournissant des outils mentaux à usage général sont même avancés. Par rapport à ces « revendications exagérées (*exaggerated claims*) », (Tedre et Denning, 2016) passent en revue une série d'études qui montrent que la pensée informatique ne confère pas des savoir-faire, dans la résolution des problèmes, transférables à d'autres domaines non informatiques, et que donc elle semble être limitée à l'informatique.

Certains de ces arguments sont repris par le rapport de l'Académie des sciences (AdS, 2013). On insiste que nous vivons dans un monde numérique, marqué par « l'impact considérable de l'informatique dans un nombre toujours croissant de domaines de l'industrie, de la communication, des loisirs, de la culture, de la santé, des sciences et de la société en général » (AdS, 2013, p. 4). De plus, « de par l'universalité de son objet, la science informatique interagit de façon étroite avec pratiquement toutes les autres sciences. Elle ne sert plus seulement d'auxiliaire de calcul, mais apporte des façons de penser nouvelles. [...] L'enseignement [de la SI] doit s'adresser [...] à tous les citoyens, pour qu'ils comprennent les mécanismes et façons de penser du monde numérique qui les entoure et dont ils dépendent » (AdS, 2013, p. 4, 5). L'argumentaire du changement des représentations mentales des outils et des objets, influencées par la SI, est donc particulièrement bien présent en France.

Une autre facette, plus économique, de l'argumentaire en faveur de l'enseignement de l'informatique est également avancée. L'informatique est ainsi qualifiée d'une importance toujours grandissante en termes de création de richesses et d'emplois dans le monde (AdS, 2013, p. 4). La pression de l'industrie en manque de personnel bien formé en informatique (AdS, 2013, p. 4) est mise en avant. Sur le marché global il s'agit de rester compétitif, et « l'Europe et la France en particulier accusent un important retard conceptuel et industriel dans le domaine par rapport aux pays les plus dynamiques » (AdS, 2013, p. 4). Pour rattraper ce retard, il faut « basculer notre pays de l'état de consommateur de ce qui est fait ailleurs à celui de créateur du monde de demain » (AdS, 2013, p. 4).

Deux types d'arguments sont ainsi principalement engagés dans la justification de l'entrée de la SI dans l'enseignement de l'informatique à l'école, l'un économique, l'autre lié aux représentations évolutives qui incorporent des savoirs informatiques. Si le premier, économique, concerne avant tout les spécialistes, les futurs chercheurs et professionnels de la SI en tant que principaux concepteurs des futurs produits commerciaux, l'autre, lié au changement de représentations, touche un public plus large. Si ces arguments sont de premiers candidats pour justifier l'entrée de la SI dans les programmes scolaires, ils ne peuvent être les seuls fondements justificatifs.

En effet, si l'argumentaire lié aux représentations évolutives incorporant des savoirs informatiques est susceptible d'être porté par la vision de l'élève en tant que citoyen en devenir (Chevallard et Wozniak, 2003), le second, économique, apparaît moins stable ou à tout le moins, particulièrement sensible à la conjoncture.

Parce que la question est celle de la recherche des raisons d'être du développement d'un enseignement pérenne de la SI fidèle à la discipline telle qu'elle existe hors de l'école et pertinente pour la formation scolaire des élèves, nous tentons la construction d'une troisième voie argumentative, en empruntant une démarche similaire à Chevallard et Wozniak (2003).

La réalité extra-informatique : l'automatisme / la répétabilité ?

Dans sa recherche des raisons d'être de l'enseignement de la statistique dans le secondaire, Chevallard et Wozniak (2003) identifient la variabilité comme une caractéristique essentielle à la démarche statistique de recueil, de traitement et d'interprétation des données. C'est la reconnaissance de cette variabilité, réalité du monde social et naturel, qui apparaît alors nécessaire et problématique : « nous vivons dans le probable, et nos réflexes, profondément enracinés, sont ceux du certain [...] Ainsi dans tous les pays la pensée statistique a-t-elle du mal à trouver la place qu'elle mérite. » (Schwartz D., 1994, cité par Chevallard et Wozniak). Ainsi, « le cas [...] de la statistique, est quelque peu autre et pourrait être autrement révélateur que les cas, de longue date naturalisés, de la géométrie ou de l'arithmétique. Pour ces domaines d'intervention de la raison mathématique, en effet, il existe chaque fois une réalité extra-mathématique relativement hypostasiée dans la culture courante, la "spatialité" pour la géométrie, la "numérosité" pour l'arithmétique » (Chevallard et Wozniak, 2003, p. 9).

Qu'en est-il de l'informatique ? La question est liée à la définition même de l'informatique. Est-elle une science, et/ou (exclusivement) une ingénierie, aucune de ces deux (et donc quelque chose complètement nouvelle – étude ? philosophie ? art ?), les deux à la fois ? Est-elle à définir en lien avec les ordinateurs (ingénierie des outils), ou plutôt avec la calculabilité (science ou mathématique des algorithmes), ou encore avec l'information ? Des questions fondamentales pour la philosophie de l'informatique (Rapaport, 2018), auxquelles des réponses différentes ont été apportées depuis les années 1950. Parmi tous ces questionnements, nous nous concentrons ici sur ce qui a été identifié comme « l'une des questions philosophiques et pratiques les plus inspirantes de la civilisation contemporaine : Qu'est-ce qu'on peut automatiser ? (*The question "What can be automated?" is one of the most inspiring philosophical and practical questions of contemporary civilization*) » (Forsythe, 1969, p. 92), cité par (Knuth, 1972).

Cette question centrale : « Qu'est-ce qu'on peut calculer (et comment) ? (*What can be computed (and how) ?*) » (Rapaport, 2017) a été reprise par de nombreux informaticiens, par exemple (Arden, 1983), (Denning, 1985). Elle est à la base de quatre

autres questions concomitantes (Rapaport, 2017) lesquelles ont irrigué l'histoire de l'informatique :

- « Qu'est-ce que l'on peut calculer de manière efficace, et comment ? (*What can be computed efficiently, and how ?*) », par exemple Comer (1989), qui est la question centrale de la théorie du calcul, avec le fameux problème ouvert $P = NP ?$;
- « Qu'est que l'on peut calculer concrètement, en pratique, et comment ? (*What can be computed practically, and how ?*) », laquelle tente de déterminer des heuristiques, des solutions « proches » des optima ;
- « Qu'est-ce que l'on peut calculer physiquement, et comment ? (*What can be computed physically, and how ?*) », qui s'intéresse à l'exécution d'une fonction calculable sur un support physique, réel, un ordinateur, et à l'investigation des implémentations physiques alternatives pour la calculabilité (quantum, DNA, etc.) ;
- « Qu'est que l'on devrait calculer, et comment ? (*What should be computed, and how ?*) », par exemple Arden (1983) et Tedre (2006), qui s'intéressent aux questions éthiques sur les décisions possiblement à laisser aux algorithmes (c'est par exemple le cas récent des voitures autonomes) et le cas échéant à la construction des intelligences artificielles : devrait-on le faire et si oui, quelles seraient nos obligations éthiques envers eux et vice versa...

Ainsi, s'intéresser, sous ces différentes formes, à ce que l'on est susceptible de calculer/automatiser apparaît comme une question centrale de l'informatique. Mais cette question en appelle très directement une seconde : qu'est-ce qu'automatiser, que cela recouvre-t-il ? comment en penser une définition opératoire ?

Dans le domaine de la psychologie, les chercheurs ont étudié deux types de processus mentaux, connus sous une multiplicité de dénominations : « conscient/délibéré – non-conscient, contrôlé – automatique, explicite – implicite, systématique – heuristique (*conscious-nonconscious, controlled-automatic, explicit-implicit, systematic-heuristic*) » (Bargh et Chartrand, 1999). Le premier est « lourd (*heavy*) » : réguler consciemment et volontairement son comportement, ses évaluations, ses décisions et ses états émotionnels exige un effort considérable et est relativement lent. Le deuxième est « léger (*light*) » : les processus non-conscients ou automatiques sont non-intentionnels, aisés, très rapides (Bargh et Chartrand, 1999).

Dans un article de synthèse de référence sur l'automatisme (*automaticity*), Bargh et Chartrand (1999), en adoptant un point de vue psychologique, introduit le concept d'automatique en tant que « quelque chose qui arrive, quoi qu'il en soit, tant que certaines conditions sont remplies (*something that happens, no matter what, as long as certain conditions are met*) ». Les auteurs font ensuite référence aux nombreux appareils et systèmes automatiques qu'on rencontre tous les jours dans nos sociétés modernes technologiques et qui sont « conçus et destinés à nous libérer des tâches qui ne nécessitent pas vraiment notre vigilance et notre intervention, afin que notre

temps et notre énergie puissent être dirigés vers celles qui le nécessitent (*devised and intended to free us from tasks that don't really require our vigilance and intervention, so that our time and energy can be directed toward those that do*) ». Ainsi, « tout comme les dispositifs mécaniques automatiques nous libèrent de l'obligation de s'occuper de et d'intervenir pour que l'effet désiré se produise, les processus mentaux automatiques libèrent la capacité limitée de notre attention consciente [...] des tâches pour lesquelles ils ne sont plus nécessaires (*just as automatic mechanical devices free us from having to attend to and intervene in order for the desired effect to occur, automatic mental processes free one's limited conscious attentional capacity [...] from tasks in which they are no longer needed*) ». En conclusion, « tout processus d'une complexité suffisante pour intéresser les psychologues sociaux implique une interaction complexe entre les processus contrôlés (conscients) et les processus automatiques (*any process of sufficient complexity to be of interest to social psychologists involves a complex interplay between both controlled (conscious) and automatic processes*) » (Bargh, 2012).

De ce point de vue, il apparaît raisonnable de considérer que l'humanité, dans sa recherche d'« allègement » de processus mentaux, vise à trouver ce qui peut être automatisé et en conséquence à construire des dispositifs pour effectuer ces tâches.

Dans notre monde numérique, l'informatique se manifeste donc avant tout dans l'automatisation de tâches répétitives (et souvent pénibles). Passer l'aspirateur, piloter des engins (avions, trains, plus récemment des voitures), pour ne citer que quelques-unes des tâches les plus connues, sont aujourd'hui passées au filtre de l'automatisation, à la recherche d'actions candidates à la répétabilité. Ainsi, l'automatisme – la répétabilité ne serait-elle pas la raison extra-informatique, la réalité du monde, l'équivalent de la variabilité dans le champ de la statistique, de la spatialité dans la géométrie et de la numérosité dans l'arithmétique ? Dès lors, l'enseignement de l'informatique devrait-il veiller à ne pas mettre de côté la question de l'identification des types de problèmes automatisables et ne pas se cantonner à la seule question de la manière de les résoudre ?

Une telle perspective ouvre ainsi la voie à de nouvelles questions à mettre à l'étude : pourquoi rendre automatisable la solution d'un problème donné ? À quelles conditions est-il raisonnable de s'engager dans une telle démarche ? Si l'on demande à un élève pourquoi sa calculatrice dispose d'un solveur d'équations, pourra-t-il (devra-t-il ?) apporter en guise de première réponse : pour ne pas avoir à résoudre les équations à la main ou au mieux (peut-être...) pour pouvoir vérifier les solutions qu'il aura obtenues autrement ? Ces premières réponses tout à fait recevables font l'économie d'une enquête sur d'autres formes de raisons, internes à la discipline informatique et renvoyant à la question de l'automatisme – la répétabilité.

Dans le cadre de la discipline informatique, la question des types de problèmes automatisables se situent dans le domaine de la théorie du calcul : la décidabilité (avec l'exemple du problème de l'arrêt non décidable), les types de problèmes P

(*polynomial*) et NP-durs (*non-deterministic polynomial*). Bien entendu, cela conduit à s'intéresser à la transposition de ces savoirs savants dans l'institution secondaire et des conditions et contraintes associées d'un tel enseignement transposé. De façon plus partielle, la question du choix des savoirs à enseigner de la SI se pose alors fortement, ainsi que celle de l'organisation didactique de son étude.

Conclusion

Dans cet article, nous avons présenté une étude exploratoire abordant la délicate question des raisons d'être de l'enseignement de l'informatique dans un contexte institutionnel de réhabilitation de cet enseignement. Nous avons voulu montrer que conduire une telle entreprise pose de nombreuses questions et autant de défis qu'il s'agit d'aborder de front. En tout état de cause, cette investigation ne peut se limiter à des raisons par ailleurs souvent avancées, économiques mais aussi relevant peu ou prou du registre de l'utilité sociale, dès lors que le but avoué d'une telle investigation est celui de penser la transposition des savoirs informatiques dans l'institution scolaire.

Bien entendu, notre étude de la question, nécessairement exploratoire, n'épuise pas le sujet. Ainsi, si l'examen de l'histoire de l'enseignement de l'informatique permet de mieux comprendre que sa réhabilitation aujourd'hui dans l'enseignement n'a rien du hasard et s'inscrit dans une forme de continuité, elle dit également que la question des raisons d'être de cet enseignement reste aujourd'hui encore entière. Sa nécessité est d'autant plus importante dès lors qu'il s'agirait d'ériger l'informatique en tant que discipline autonome, libérée, au moins partiellement¹⁵, de l'autorité de la discipline mathématiques. Nous avons proposé l'automatisme, la répétabilité comme motivation principale de l'étude de la science informatique. Mais si un tel choix est retenu, une autre forme de nécessité apparaît : celle de disposer d'un savoir de référence, d'une organisation de la science informatique en domaines et d'un choix de domaines à enseigner, portés par cette raison d'être.

In fine, ce sont toutes ces questions et bien d'autres encore qu'il reste à investir... à voir avant tout comme autant de défis pour la recherche en didactique de l'informatique !

Références

Académie des sciences (AdS). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*. 2013. <http://www.cfem.asso.fr/debats/mathematiques-informatique/rapport-de-l-academie-des-sciences-sur-l-enseignement-de-l-informatique>.

15. car il ne s'agit pas de nier l'histoire longue et commune des mathématiques et de l'informatique par ailleurs.

Archambault, J.-P. 1985, *vingt ans après... Une histoire de l'introduction des TIC dans le système éducatif français*. Medialog 54:42–45, 2005.

Archambault, J.-P. *L'informatique, outil et objet d'enseignement*. 1024 – Bulletin de la société informatique de France, 1, septembre 2013, pp. 70–79, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2013/11/1024-numero-1-archambault.pdf>.

Archambault, J.-P. *Le plan informatique pour tous dans l'académie de Créteil : une enquête d'évaluation*. 1024 – Bulletin de la société informatique de France, 6, juillet 2015, pp. 97–104, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2015/07/1024-no6-archambault.pdf>.

Archambault, J.-P. *Un enseignement de l'informatique de culture générale pour tous les élèves – Pourquoi et comment*. 1024 – Bulletin de la société informatique de France, 11, septembre 2017, pp. 61–76, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2017/10/1024-no11-Archambault.pdf>.

Arden, B. W. *What can be Automated? : Computer Science and Engineering Research Study*. MIT Press, Cambridge, MA, USA, 1983.

Artigue, M. et Raoult, J.-P. *À propos du rapport de l'Académie des sciences sur l'enseignement de l'informatique*. 2013. <http://images.math.cnrs.fr/A-propos-du-rapport-de-l-Academie.html>.

Bargh, J. A., and Chartrand, T. L. *The unbearable automaticity of being*. American psychologist 54, no. 7:462, 1999.

Bargh, J. A., Schwader, K. L., Hailey, S. E., Dyer, R. L., and Boothby, E. J. *Automaticity in social-cognitive processes*. Trends in cognitive sciences, 16(12), 593–605, 2012.

Baron, G.-L. et Drot-Delange, B. et Grandbastien, M. et Tort, F. *Computer science education in french secondary schools : Historical and didactical perspectives*. Trans. Comput. Educ, 14(2):27, 2014.

Baron, G.-L. et Drot-Delange, B. *L'éducation à l'informatique à l'école primaire*. 1024 – Bulletin de la société informatique de France, 9, novembre 2016, pp. 73–79, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2016/11/1024-no9-Baron-Drot-Delange.pdf>.

Baudé, J. *Association francophone pour la didactique de l'informatique (AFDI, 1988-1996)*. 1024 – Bulletin de la société informatique de France, 9, novembre 2016, pp. 81–93, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2016/11/1024-no9-AFDI.pdf>.

Bell, T., Andreae, P. and Robins, A. *A Case Study of the Introduction of Computer Science in NZ Schools*. Trans. Comput. Educ., 14(2):10:1–10:31, 2014. ISSN 1946-6226.

du Boulay, B. *Some difficulties of learning to program*. Journal of Educational Computing Research, 2(1):57–73, 1986.

Brown, N. C. C., Sentance, S., Crick, T. and Humphreys, S. *Restart : The Resurgence of Computer Science in UK Schools*. Trans. Comput. Educ., 14(2):9:1–9:22, 2014. ISSN 1946-6226.

- Brulé, J.-P. *L'informatique, malade de l'État : du Plan calcul à Bull nationalisée, un fiasco de 40 milliards*. Vol. 1. Belles Lettres, 1993.
- Chevallard, Y. et Wozniak, F. *Enseigner la statistique au secondaire. Entre genre prochain et différence spécifique*. Actes de la XII^e école d'été de didactique des mathématiques, 2003.
- Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J. and Young, P. R. *Computing as a discipline*. Commun. ACM 32, 1, 9–23, 1989.
- Denning, P. J. *What is computer science ?*. Am. Sci. 73, 1, 16–19, 1985.
- Douady, R. *Jeux de cadres et dialectique outil-objet*. Recherches en Didactique des Mathématiques, 7(2), 1986.
- Forsythe, G. E. *What to do till the computer scientist comes*. The American Mathematical Monthly 75, no. 5:454–462, 1968.
- Forsythe, G. E. *Computer Science and Education*. Proceedings of IFIP Congress 1968. August 5th-10th 1968, Edinburgh, UK : pp. 92–106, 1969, (Volume 2).
- Gal-Ezer, J. and Stephenson, C. *A Tale of Two Countries : Successes and Challenges in K-12 Computer Science Education in Israel and the United States*. Trans. Comput. Educ., 14(2):8:1–8:18, 2014. ISSN 1946–6226.
- de la Higuera, C. *La didactique de l'informatique. Entretien avec Étienne Vandeput*. 1024 – Bulletin de la société informatique de France, 8, avril 2016, pp. 105–111, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2016/04/1024-no8-Vandeput.pdf>.
- Knuth, D. E. *George Forsythe and the development of computer science*. Commun. ACM 15, 8, 721–726, 1972.
- Ministère de l'Éducation nationale de l'Enseignement supérieur et de la Recherche (MENSUR). *Programmes d'enseignement du cycle des apprentissages fondamentaux (cycle 2), du cycle de consolidation (cycle 3) et du cycle des approfondissements (cycle 4)*. Bulletin officiel spécial no. 11 du 26 novembre 2015. http://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=94717
- Nijimbere, C et Haspekian, M. *Les enseignants face à l'entrée de l'algorithmique dans l'enseignement des mathématiques au lycée scientifique en France*. CORFEM : Actes des 17e et 18e colloques, pp. 265–285, 2012.
- Nijimbere, C. *L'enseignement de savoirs informatiques pour débutants, du second cycle de la scolarité secondaire scientifique à l'université en France : une étude comparative*. Thèse de doctorat. Université Sorbonne Paris Cité, 2015.
- Rapaport, W. J. *What Is Computer Science*. University at Buffalo, The State University, 2017.
- Rapaport, W. J. *Philosophy of Computer Science*, DRAFT ©2004–2017 by William J. Rapaport, January 14, 2018, <https://www.cse.buffalo.edu/~rapaport/Papers/phics.pdf>.
- Rogalski, J. *Psychologie de la programmation, didactique de l'informatique : déjà une histoire*. In G.-L. Baron, E. Bruillard, E., & B. Drot-Delange (éds.). *L'information en éducation : perspectives curriculaires et didactiques*, pp. 279–305, 2015.

Rogalski, J. et Samurçay, R. *Les problèmes cognitifs rencontrés par des élèves de l'enseignement secondaire dans l'apprentissage de l'informatique*. European Journal of Psychology of Education, 1(2):97–110, 1986.

Rouchier, A. *Objets de savoir de nature informatique dans l'enseignement secondaire*. 1990.

Samurçay, R. *Signification et fonctionnement du concept de variable informatique chez des élèves débutants*. Educational Studies in Mathematics, 16(2):143–161, 1985.

Schwartz, D. *Le jeu de la science et du hasard. La statistique et le vivant*. Paris, Flammarion, 1994.

Simon, J.-C. *L'éducation et l'informatisation de la société : rapport au président de la république*. La Documentation française, 1980.

Société informatique de France (SIF). *Enseigner l'informatique de la maternelle à la terminale*. 1024 – Bulletin de la société informatique de France, 9, novembre 2016, pp. 25–33, <https://www.societe-informatique-de-france.fr/wp-content/uploads/2016/11/1024-no9-enseigner-informatique.pdf>.

Sorva, J. *A roles-based approach to variable-oriented programming*. 2008.

Tedre, M. *What should be automated? : The fundamental question underlying human-centered computing*. In Proceedings of the 1st ACM international workshop on Human-centered multimedia (HCM '06). ACM, New York, NY, USA, 19–24, 2006.

Tedre, M. and Denning, P. J. *The long quest for computational thinking*. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16). ACM, New York, NY, USA, 120–129, 2016.

Wing, J.-M. *Computational thinking*. Commun. ACM, 49(3):33–35, March 2006. ISSN 0001-0782.

