

Enjeux de durabilité et problématisation en informatique

Baptiste de Goer^{1,3}, Micha Hersch² et Sophie Quinton¹

¹ Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

² Haute Ecole Pédagogique Vaud, 1007 Lausanne, Suisse

³ Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, F-38000 Grenoble, France

Résumé. Cet article explore les difficultés d’articulation entre informatique scolaire et éducation à la durabilité. En appliquant la théorie de la problématisation de Fabre, nous montrons que l’informatique, souvent orientée vers la résolution de problèmes, accorde une place limitée à la construction de ces problèmes. Les sciences humaines et sociales nous apprennent pourtant que la construction du problème est essentielle à l’étude des enjeux de durabilité liés au numérique. En nous appuyant sur ce constat, nous illustrons comment les cours d’informatique, en revalorisant la construction des problèmes dont ils traitent, pourraient évoluer vers une discipline davantage connectée aux enjeux de durabilité actuels.

Mots-clé: problématisation · didactique · informatique et société

1 Contexte et motivation

L’école joue un rôle essentiel pour former les citoyen·nes de demain aux grands enjeux de durabilité contemporains [36]. Ainsi, ces questions s’invitent dans les cursus scolaires : en plus de savoirs fondamentaux déjà enseignés dans différents cours (géographie, sciences de la vie et de la terre, ...), toutes les disciplines sont amenées à se positionner vis-à-vis de leur contribution possible à une éducation à la durabilité. L’informatique n’y échappe pas, alors que les technologies numériques qui en sont issues sont au cœur de controverses environnementales, par exemple au regard de la consommation en ressources des centres de données [8, 5].

En classe, les enjeux de durabilité liés au numérique restent difficiles à articuler avec l’apprentissage de l’informatique [21]. Généralement, ce sujet est soit traité de manière décorrélée de la discipline, soit abordé sous le prisme des solutions que peut apporter l’informatique [40], qui se retrouve alors déresponsabilisée des conséquences de la production des technologies.

Dans cet article, nous argumentons que cette difficulté d’articulation peut s’expliquer par les modes de problématisation spécifiques à l’informatique qui ne permettent pas d’appréhender les liens complexes entre technologies et société. Cette analyse nous amène à proposer une nouvelle articulation entre éducation à la durabilité et enseignement de l’informatique qui passe par un élargissement du cadre habituel de problématisation en informatique pour soutenir une éducation

interdisciplinaire au politique. Une telle approche contribuerait à faire évoluer l’informatique scolaire vers une discipline davantage connectée aux grands enjeux soulevés par la crise environnementale et la numérisation du monde.

Le reste de l’article est organisé comme suit. La partie 2 présente les concepts théoriques mobilisés. La partie 3 souligne que la phase de construction des problèmes est négligée en informatique. La partie 4 montre que la construction du problème doit être revalorisée pour analyser les enjeux de durabilité liés au numérique. La partie 5 propose de placer l’informatique scolaire dans une démarche interdisciplinaire d’éducation au politique. La partie 6 conclut.

2 Fondements théoriques

2.1 Théorie de la problématisation

Les théories didactiques par la problématisation sont nées en France dans les années 1980 afin de dépasser, dans une approche constructiviste, le cadre de l’apprentissage par résolution de problèmes [13]. Elles sont basées sur la constatation, déjà faite par Bachelard et Kuhn, que les problèmes scientifiques ne se posent pas d’eux-mêmes. Au contraire, la construction des problèmes et la définition des moyens légitimes de les résoudre est au cœur des pratiques scientifiques. Appliqué aux disciplines scolaires, ce constat amène à revaloriser la construction des problèmes, et à faire de l’activité de *problématisation* par les élèves un cadre d’apprentissage.

D’après Fabre [16], problématiser consiste à faire interagir, dans un cadre donné, trois aspects d’un problème : sa position, sa construction et sa résolution, comme représenté sur la figure 1. La position du problème est son origine, la situation à partir de laquelle le problème émerge. La construction du problème est une démarche qui sélectionne et articule des données et des conditions pour déterminer la forme de la solution. Quant au cadre, c’est “*l’instance normative qui détermine ce à quoi il convient de donner le statut de donnée, de condition ou de solution*” [16, p.36]. Fabre mobilise l’exemple de Phileas Fogg, le héros du *tour du monde en quatre-vingt jours*, qui doit planifier un itinéraire de voyage, avec pour condition de finir le tour du monde dans les temps, et pour données les horaires des trains. Le contexte de son pari qui fait office de cadre.

Des problèmes cartésiens aux problèmes pernicioeux S’il avait été formalisé dans cette optique, le problème de Phileas Fogg aurait typiquement pu être résolu par un algorithme. Il s’agit d’un *problème cartésien*, c’est-à-dire “*a) parfaitement déterminable ; b) soluble ; c) ceci, selon une démarche de résolution explicitable [...] ; d) sa ou ses solutions se révélant évaluables selon des critères objectifs*” [17]. Tous les problèmes ne sont pas cartésiens et Fabre [17] retrace la découverte de plusieurs autres types de problèmes. Il commence par Simon qui, en essayant de développer un programme universel de résolution de problème, s’est rendu compte que certains étaient *mal structurés* (*ill-structured*), comme la composition musicale, car ils dépendent de critères subjectifs (esthétiques par

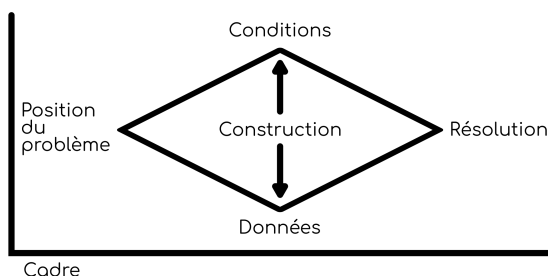


Fig. 1. Losange de problématisation tiré de [16, p.30]

exemple). Simon garde malgré tout l'espoir de réussir à les découper en sous-problèmes cartésiens. Cet idéal vacille avec les *problèmes flous*, dans lesquels plusieurs parties prenantes ayant des valeurs divergentes entrent en jeu. Quand ils deviennent inextricables à cause de situations complexes et d'enjeux élevés, il est alors possible de parler de *problèmes pernicioeux* (*wicked problems*). Ceux-ci ont été caractérisés par Rittel et Webber [42] en opposition à l'idéal, porté par Simon, de rationalisation des décisions politiques (alors perçues comme de simples problèmes mal structurés). Les catégories de problèmes mal structurés, flous et pernicioeux ne sont pas étanches, mais se différencient toutes des problèmes cartésiens.

2.2 Philosophie de l'informatique

La nature de l'informatique est, depuis son apparition, sujette à de nombreux débats [51]. Nous nous appuyons ici sur les travaux en philosophie de l'informatique de Turner⁴ [2, 49]. D'après lui, l'informatique s'intéresse à des *artéfacts numériques* (algorithme, programme, etc..) qui s'agencent en niveaux d'abstraction par-dessus une machine calculatoire physique (un ordinateur).

Ces artéfacts sont conçus dans le but de répondre à des fonctions spécifiques [2]. La *conception* transforme les fonctions en structure, puis l'implémentation transforme la structure en objet. Par exemple la fonction d'un algorithme de tri est de trier des listes. Sa structure variera entre un tri fusion, par insertion, etc.. puis sera implémentée en programme. Chaque niveau d'abstraction sert de fonction pour les niveaux inférieurs et de structure pour les niveaux supérieurs, comme présenté en table 1.

L'informatique est ainsi caractérisée par un "*régime de la commande*" [50] dans lequel, à chaque niveau d'abstraction, la fonction s'impose comme commande venue des niveaux supérieurs, depuis la commande des client-es, jusqu'aux instructions données en assembleur. La multiplication des niveaux, couplée au principe d'encapsulation, mène à "*une dissociation croissante du fonctionnement et des usages*" [1, p.115] dans laquelle la partie émettrice de la commande n'a

⁴ Il nous a semblé que Turner travaille à un niveau théorique pertinent pour être articulé avec les théories de la problématisation que nous mobilisons dans l'article.

Tableau 1. Exemples d’artéfacts numériques à divers niveaux d’abstraction, interprété de [2, 49].

	Regard porté sur les artéfacts numériques			
	Fonction Commande	Conception →	Structure	Implémentation → Artéfact
Niveaux d’abstraction	Intention client-e		Cahier des charges	Diagramme UML
	Cahier des charges		Diagramme UML	Interfaces des modules
	Diagramme UML		Interfaces des modules	Algorithmes
	Interface de module		Algorithme	Programme
	Algorithme		Programme	Code assembleur
	⋮		⋮	⋮

pas à se soucier de la manière dont cette commande est exécutée au delà des spécifications d’entrées et de sorties.

Dans cet article, nous appellerons *technologie (numérique)* l’objet résultant de l’implémentation de tous les niveaux d’abstraction en vue d’une utilisation finale (un logiciel par exemple). L’*informatique* sera quant à elle réduite à la partie de la discipline centrée autour de la conception d’artéfacts numériques.

3 La problématisation en informatique

Si le cadre d’apprentissage par problématisation a essaimé dans plusieurs disciplines scolaires [13] telles que les sciences de la vie et de la terre, l’éducation physique, l’histoire ou la littérature, il reste encore peu mobilisé en informatique. Historiquement, l’informatique s’est plutôt construite sur la résolution de problèmes⁵, qui a pris une importance croissante dans sa didactisation, pour en devenir un point central à partir des années 90 [47]. L’enseignement de l’informatique se structure ainsi souvent autour de la conception d’artéfacts numériques (par exemple par des démarches projet ou des exercices d’algorithmie) comme solutions à différents problèmes, comme les montrent les recommandations pédagogiques dans les programmes scolaires [34, 35]. Cette focalisation sur la solution hérite directement d’une transposition des pratiques de référence. L’informatique commerciale par exemple vend des « solutions » sous forme de logiciels, ou d’infrastructures.

Résoudre des problèmes par la technologie suppose pourtant de savoir les construire en terme de traitement de l’information⁶. Par exemple Standl dans [46]

⁵ Ceci est bien illustré par la citation suivante issue des actes d’une importante conférence de génie logiciel. “*The requirement - that is, the problem - is in the world; the machine is the solution we construct*” [26, p. 283]

⁶ Objectif pédagogique revendiqué des programmes scolaires, cf [34, 35]

fait concevoir à ses élèves un algorithme (la solution) pour optimiser le rangement de cartons de déménagement (le problème). Ce problème n'est pas formalisé en amont de l'exercice et les élèves sont tenu-es dans un premier temps de l'abstraire et de le décomposer afin que ses instances soient solubles par un algorithme, ce qui les amène à représenter les objets à ranger dans les cartons par leur forme géométrique. La construction du problème en informatique correspond donc à la formalisation de la commande en une spécification adéquate [23, 27] : définition formelle d'un algorithme, interface d'un module, base de données labellisées pour l'apprentissage automatique, etc.

C'est à ce moment que vont être déterminées les méthodes de validation de la solution (preuve formelle, tests, ...) et les critères associés (correction, précision, efficacité, utilisabilité, ...). À un niveau d'abstraction donné, l'informaticien-ne vérifie sa solution vis à vis de ces critères, et ne remet plus en question la commande ni sa formalisation. Ce constat a été documenté autant en algorithmique [23], qu'en apprentissage automatique [27, p.49-50], ou en interaction humain-e-machine [41]. Standl [46], ayant valorisé la construction du problème avec ses élèves, se retrouve également à dépasser ces conventions de vérification. Les élèves testent l'algorithme en conditions réelles de déménagement, et constatent qu'il nécessite trop de cartons. C'est une invitation à re-construire le problème abstrait en prenant en compte cette limite.

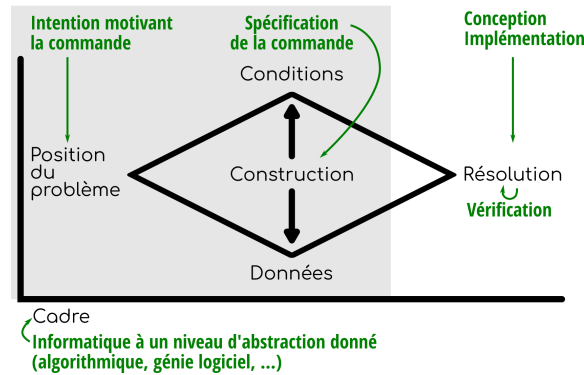


Fig. 2. Losange de problématisation (figure 1) annoté en vert pour l'informatique. En grisé la partie négligée dans le processus.

Comme illustré par la figure 2, en informatique, la construction du problème est donc négligée à deux moments. Déjà en amont de la conception, quand elle est vue comme triviale ou extérieure au travail de l'élève ou de l'informaticien-ne. En apprentissage automatique par exemple [27], la base de données labellisées correspond à la spécification de ce qui est attendu (la commande) du modèle (l'artéfact numérique), pourtant sa construction est reléguée au travail du clic, un travail considéré comme peu qualifié et hors de l'informatique [10]. Ensuite,

en aval de la conception, quand les modalités de vérification de la solution ne permettent pas de remettre en question la spécification de la commande. En apprentissage automatique, l'exactitude du modèle est vérifiée vis-à-vis de la base de données elle-même. C'est pourtant au moment de la construction du problème que se réfléchissent les tenants et aboutissants du développement d'un artéfact numérique [23, 32, 27, 3].

4 Problématiser autrement en informatique

On va le voir, le manque d'attention porté à la problématisation en informatique explique en partie les difficultés de la discipline, académique comme scolaire, à prendre en compte les enjeux de durabilité liés au numérique.

4.1 Vers une informatique consciente des conséquences de ses choix

Construire un problème de conception d'artéfact numérique suppose de faire des choix, qu'une attention renouvelée à la construction d'un tel problème permet de rendre visibles. Ainsi, dans l'exemple de Standl précédemment cité [46], la correction propose d'abstraire les objets à ranger dans les cartons de déménagement par leur forme géométrique, et ne prend pas en compte leur poids. Or, transporter des cartons de déménagement lourds peut être facteur de blessures, tandis que prendre en compte ce risque aurait pour conséquence de complexifier l'algorithme. Cet exemple illustre comment les choix dans la construction du problème se font selon des critères motivés par des valeurs qui peuvent être de différentes natures (sociales, cognitives, épistémiques) [25].

De plus, ces choix conditionnent la manière dont l'artéfact va être implémenté puis utilisé [20, 32], et donc ses effets sur les personnes et l'environnement. Par exemple, qu'un site internet soit utilisable par une personne non-voyante relève de choix de conception qui participent à éviter l'exclusion d'une partie de la population. Or, au-delà des informaticien·nes, il peut y avoir des divergences de valeurs et d'intérêts entre les différentes parties prenantes concernées par les conséquences des choix de conception, notamment les personnes utilisatrices de l'artéfact, les personnes à l'origine de la commande, mais aussi les personnes en aval de la commande [20, 32]. Par exemple, des entreprises de services numériques (réseaux sociaux, plateformes de vente en ligne, ...) sont régulièrement condamnées par la justice pour leur usage de *dark patterns* [7], des éléments d'interface conçus dans le but de tromper les utilisateurs et utilisatrices. Les choix de construction du problème se font alors en faveur des un·es et au détriment des autres et sont, en ce sens, intrinsèquement politiques. Un tel critère de conception ne s'inscrit pourtant pas dans la problématisation traditionnelle ayant cours en informatique.

4.2 Problématisation et enjeux de durabilité liés au numérique

Dans le cas des enjeux de durabilité liés au numérique, les choix de conception peuvent avoir de multiples conséquences bien documentées. Par exemple,

concevoir un logiciel de lecture de musique basé intégralement sur du streaming plutôt que sur une gestion en local de fichiers musicaux permet de mutualiser le stockage des fichiers audio et nécessite moins de mémoire en local. En revanche, un tel choix de conception nécessite une connexion internet en continu et de bonne qualité, et donc l'existence d'infrastructures cloud adaptées (réseaux et centres de données). Le cycle de vie de ces infrastructures a des conséquences en termes de consommation de ressources matérielles, d'approvisionnement électrique, de pollutions, d'emprise au sol, etc. [15, 33, 9], qui s'accompagnent d'impacts sur les populations proches des sites de production, d'implantation et/ou de traitement de la fin de vie de ces infrastructures. Par ailleurs, le streaming favorise des types spécifiques de modèles économiques, reconfigure notre rapport à la consommation de musique [38], rend possible l'analyse fine des comportements à des fins publicitaires qui ont elles-mêmes des conséquences environnementales, etc..

On comprend aisément sur l'exemple du streaming à quel point les liens entre la conception d'un artéfact numérique et ses conséquences en termes de durabilité sont complexes. En effet, les problèmes liés aux enjeux de durabilité ne sont pour la plupart pas de nature cartésienne : ils ont de multiples dimensions politiques [28, 22], qu'il n'est ni souhaitable ni possible de réduire à des problèmes cartésiens de traitement de l'information [14, 17]. De plus, les enjeux globaux de durabilité sont eux-mêmes des problèmes pernicioeux, qui peuvent être abordés de différentes manières. Par exemple, protéger l'environnement pour lui-même ou parce qu'il est utile à l'humanité sont deux manières très différentes d'approcher la question. Il en est de même pour le lien entre numérique et durabilité : voir les technologies numériques comme des outils au service d'un monde durable n'amène pas aux mêmes conclusions que les considérer comme des accélératrices de la surconsommation. Ces différences de point de vue relèvent de litiges [16], c'est-à-dire de désaccords de choix de cadres de problématisation des enjeux de durabilité liés au numérique, et rajoutent des couches de perniciosité à ces problèmes [22]. Il est à noter que cette difficulté dans le cadrage du problème semble être au moins partiellement un impensé dans la communauté de recherche qui s'intéresse aux enjeux de durabilité liés au numérique [22].

En résumé, concevoir ou évaluer un artéfact numérique à l'aune des questions de durabilité est un problème pernicioeux, qu'il n'est pas possible de problématiser sans sortir du cadre traditionnel de l'informatique.

4.3 Vers une informatique consciente de ses limites

On vient de le voir, comprendre pleinement les intrications entre technologies et société implique de sortir d'un cadre de problématisation visant à concevoir des artéfacts numériques, pour adopter un cadre permettant de les étudier dans toute leur conflictualité, leur complexité, leur polysémie (plusieurs interprétations possibles), leur ouverture (plusieurs solutions possibles) et leur temporalité critique (urgence des actions) [17].

En sciences sociales, les approches *Science, Technique et Société* (STS) [29] fournissent des cadres de problématisation adaptés à cette exploration, par ex-

emple en analysant des controverses socio-techniques sous le prisme des “*inter-relations entre des techniques, des pratiques, des concepts, des groupes sociaux et des politiques*” [45]. Pour les enjeux de durabilité liés au numérique, les STS ont notamment analysé les promesses, pratiques et tensions qui émergent des diverses tentatives de mise en œuvre du numérique au service de la durabilité [39] : visualisation de données environnementales, agriculture de précision, plateformes coopératives, ... Elles se sont également attachées à visibiliser le fonctionnement matériel du numérique [11], par exemple en documentant comment les infrastructures réseau (antennes, centres de données, ...) façonnent l’espace public à travers leur histoire, les discours qui justifient leur croissance, les conflits de ressources qu’elles provoquent, les mobilisations que leur implantation génèrent, etc...

En termes de problématisation, les STS se concentrent sur la construction du problème, en particulier parce que ce type de problème n’est pas strictement résoluble. Il peut tout au plus être déplacé, atténué ou transformé, et toute solution ne pourra être que partielle, satisfaisante pour certaines parties prenantes, mais pas pour d’autres [18]. Du point de vue politique, il est cependant possible de se positionner sur ces problèmes et d’agir. Les modalités d’action des informaticien·nes peuvent parfois relever de la conception de nouveaux artefacts numériques à condition que cela soit justifié par la construction du problème, tout en ayant conscience des limites de la technique, et en assumant les valeurs qui motivent les choix de conception et leurs conséquences. Certains domaines de la recherche en conception (*design*) comme le *value sensitive design* [19], ou certaines branches du *sustainable HCI* [6], s’attellent à identifier des méthodes pour le faire. Les modalités d’action en tant qu’informaticien·nes ne relèvent cependant pas nécessairement de la fabrication de nouvelles technologies, d’autres exemples ont été documentées [44] : penser la réparation des technologies, leur résilience, leur décroissance, etc.

5 Quelles conséquences pour la didactique ?

Nous avons vu que les enjeux de durabilité liés au numérique sont des problèmes pernicioeux. Les traiter en classe relève donc d’une éducation scolaire au politique [12], dans la lignée plus générale de l’éducation à la durabilité [37, 4]. Cette éducation vise à développer chez l’élève ses propres capacités d’analyse et de positionnement sur des questions de société, sans chercher à inculquer aux élèves des opinions politiques prédéfinies [18]. Par exemple, Barthes [4] propose d’apprendre aux élèves à analyser l’origine des connaissances en jeu, les différentes positions dans une controverse, l’intrication des problèmes entre eux, les rapports de pouvoir entre les parties impliquées et les injustices présentes.

D’après Fabre, les méthodes didactiques par la problématisation centrées sur des cas d’étude sont à privilégier dans ce contexte [18]. Chauvigné et Fabre [12] s’inspirent d’une méthode pédagogique issue des STS, la cartographie de controverses [45], mais il en existe d’autres comme le débat [48] ou les démarches

d'enquête interdisciplinaires [24, 31], un paradigme pédagogique dans lequel les élèves forment un public qui enquête librement sur le problème.

Dans tous les cas, en tant que problème socio-scientifique, les connaissances nécessaires à la problématisation des enjeux de durabilité liés au numérique relèvent de disciplines variées [43]. Bien positionnée comme support à une éducation interdisciplinaire au politique, l'informatique permet d'éclairer les enjeux politiques autour d'une technologie sans pour autant se positionner comme solution. Elle peut d'une part visibiliser le fonctionnement des technologies, et d'autre part questionner les choix techniques effectués au regard des valeurs qui les motivent.

Reprenons comme cas d'étude les enjeux de durabilité liés au développement d'une application de streaming musical. Dans une perspective d'éducation au politique, les élèves pourraient cartographier des controverses autour de l'implantation de centres de données nécessaires au stockage des fichiers musicaux, ou mener des débats sur les enjeux de la numérisation des œuvres musicales. Des connaissances en informatique sont ici utiles pour visibiliser en quoi des centres de données sont nécessaires au fonctionnement de l'application, ou le fait qu'il existe des méthodes d'encodage des fichiers musicaux visant à limiter le téléchargement illégal (le *digital right management*). Divers choix techniques dans l'application peuvent être questionnés au regard des valeurs qui les motivent : pourquoi développer une nouvelle application plutôt qu'utiliser un lecteur cassette ? Quels sont les enjeux de centraliser ou décentraliser le stockage des fichiers musicaux ? Vaut-il mieux utiliser un stockage cloud ou un serveur auto-hébergé ? Etc. On voit bien ici l'articulation entre savoirs disciplinaires et éducation au politique autour d'enjeux de durabilité liés au numérique.

6 Conclusion

Aborder en classe les enjeux de durabilité liés au numérique en prenant en compte leur nature pernicieuse est un défi pour l'informatique scolaire. Dans cet article, nous avons proposé d'approcher ce défi en mettant l'informatique scolaire au service d'une éducation interdisciplinaire au politique. Lors de tels cours, les élèves apprendraient, au moment de développer des outils, à assumer les valeurs motivant leurs choix, dans la lignée du *value sensitive design* [19] et des approches critiques [30]. Lors de l'analyse des outils développés, les élèves apprendraient à en comprendre les conséquences réelles au delà de l'adéquation avec la spécification [32, 23, 3].

Dans ce contexte, l'informatique scolaire évoluerait d'une didactique de la résolution de problèmes vers une didactique permettant aux élèves de problématiser la diversité des choix techniques et leurs conséquences. La pertinence d'une telle évolution dépasse par ailleurs les enjeux de durabilité. Ainsi, dans un contexte d'explosion des usages de l'IA générative, dont les enjeux sont également sociaux, éducatifs et cognitifs, revaloriser l'étape de construction des problèmes paraît à la fois urgent et important.

En conclusion, notons qu'un tel changement pédagogique nécessiterait pour les enseignant-es de se familiariser avec le traitement en classe des valeurs, et avec le travail interdisciplinaire. Cette charge supplémentaire pose question, et il serait utile de connaître la motivation des enseignant-es à se former à ces sujets.

Remerciements. Les auteur-ices remercient le ou la relecteur-ice anonyme pour la référence à M. Jackson. M.H. remercie sa collègue Morgane Chevalier de l'avoir orienté vers la théorie de la problématisation de Fabre.

References

1. Alioui, J.: Penser le "numérique" avec Gilbert Simondon. Epistémé, Lausanne (2025)
2. Angius, N., Primiero, G., Turner, R.: The Philosophy of Computer Science. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Spring 2025 edn. (2025)
3. de Arruda Falcão, J.P., de Lemos Meira, S.R., Ramalho, G.L.: Algorithmic pragmatism: First steps. In: 2021 IEEE International Symposium on Technology and Society (ISTAS). pp. 1–8. IEEE (2021)
4. Barthes, A.: Quels curricula d'éducation au politique dans les questions environnementales et de développement? Éducation et socialisation. Les Cahiers du CER-FEE (63) (2022)
5. Bouveret, S., Bugeau, A., Orgerie, A.C., Quinton, S.: De l'eau dans les nuages. Annales des Mines - Enjeux Numériques (27), 40–47 (2024)
6. Bremer, C., Knowles, B., Friday, A.: Have we taken on too much?: A critical review of the sustainable hci landscape. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22, Association for Computing Machinery, New York, NY, USA (2022)
7. Brignull, H., Leiser, M., Santos, C., Doshi, K.: Deceptive patterns - legal cases. <https://www.deceptive.design/cases> (2025), visité le 4 décembre 2025
8. Brodie, P.: Smarter, greener extractivism: Digital infrastructures and the harnessing of new resources. Information, Communication & Society **28**(6), 1061–1080 (2025)
9. Carnino, G., Marquet, C.: Les datacenters enfoncent le cloud : enjeux politiques et impacts environnementaux d'internet. Zilsel **2018**/1(3), 19–62 (2018)
10. Casilli, A.A.: Digital Labor : travail, technologies et conflictualités. In: Qu'est-ce que le digital labor ?, pp. 10–42. Editions de l'INA (2015)
11. Cellard, L., Marquet, C.: Matérialités environnementales du numérique. RESET. Recherches en sciences sociales sur Internet (15) (2025)
12. Chauvigné, C., Fabre, M.: Des "éducations à" aux problèmes pernicioeux: repères pour une éducation au politique à l'école. Travail et Apprentissages (2), 140–159 (2023)
13. Doussot, S., Hersant, M., Lhoste, Y., Orange Ravachol, D.: Le cadre de l'apprentissage par problématisation. Presses Universitaires de Rennes (2022)
14. Easterbrook, S.: From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. In: ICT for Sustainability 2014 (ICT4S-14). pp. 235–244. Atlantis Press (2014)
15. Ensmenger, N.: The Cloud Is a Factory. In: Your Computer Is on Fire. The MIT Press (03 2021)

16. Fabre, M.: Qu'est-ce que problématiser. Vrin - Chemins Philosophiques, Paris (2017)
17. Fabre, M.: Problématologie des questions socialement vives. repères épistémologiques pour l'école. *Revue française de pédagogie* pp. 89–99 (2021)
18. Fabre, M.: Vous avez dit éducation au politique? Penser l'éducation. *Philosophie de l'éducation et histoire des idées pédagogiques* (54), 31–47 (2024)
19. Friedman, B., Hendry, D.G.: Value sensitive design: Shaping technology with moral imagination. Mit Press (2019)
20. Friedman, B., Kahn Jr, P.H.: Human values, ethics, and design. In: *The human-computer interaction handbook*, pp. 1267–1292. CRC press (2007)
21. de Goër, B., Hersch, M., Quinton, S.: Informatique et durabilité, une difficile transposition didactique. In: *DidaSTIC 2024-Colloque Didapro 10 sur la Didactique de l'informatique et des STIC*. pp. 23–32 (2024)
22. de Goër, B., Quinton, S.: The political dimension of problematizing ict-related sustainability issues. In: *2025 11th International Conference on ICT for Sustainability (ICT4S)*. pp. 274–285. IEEE (2025)
23. Green, B., Viljoen, S.: Algorithmic realism: expanding the boundaries of algorithmic thought. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. pp. 19–31 (2020)
24. Hervé, N., Lipp, A., Cancian, N., Panissal, N., Vidal, M.: Quels modèles de l'enquête sur une question socialement vive pour le développement d'une pensée problémátologique? *Questions Vives. Recherches en éducation* (37) (2022)
25. Hirsch Hadorn, G., Baumberger, C.: What types of values enter simulation validation and what are their roles? In: *Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*, pp. 961–979. Springer (2019)
26. Jackson, M.: The world and the machine. In: *Proceedings of the 17th International Conference on Software Engineering*. pp. 283–292 (1995)
27. Jaton, F.: *The Constitution of Algorithms - Ground-Truthing, Programming, Formulating*. The MIT Press (apr 2021)
28. Jaton, F., Vinck, D.: Politicizing algorithms by other means: Toward inquiries for affective dissensions. *Perspectives on Science* **31**(1), 84–118 (2023)
29. Joly, P.B., Bonneuil, C.: *Science, Techniques et Société. Repères* (Maspero), La Découverte (2013)
30. Kirdani-Ryan, M., Ko, A.J.: The house of computing: Integrating counternarratives into computer systems education. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. pp. 279–285 (2022)
31. Levinson, R.: Introducing socio-scientific inquiry-based learning (ssibl). *School Science Review* **100**(371), 31–35 (2018)
32. Lin, K.: Do abstractions have politics? toward a more critical algorithm analysis. In: *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. pp. 1–5. IEEE (2021)
33. Lopez, F.: *À bout de flux*. Editions Divergences (2022)
34. Ministère de l'Éducation nationale et de la Jeunesse: Programme de numérique et sciences informatiques de première générale (2019), <https://eduscol.education.fr/document/30007/download>
35. Ministère de l'Éducation nationale et de la Jeunesse: Programme de numérique et sciences informatiques de terminale générale (2019), <https://eduscol.education.fr/document/30010/download>

36. Ministère de l'Éducation Nationale et de la Jeunesse: Vademe-cum. Éducation au développement durable. horizon 2030. (2021), <https://eduscol.education.fr/document/5239/download>
37. Morin, O.: La formation de la pensée écologique, éducation scientifique et politique. In: Une éducation à la pensée scientifique pour une société plus juste. (2018)
38. Nowak, R., Glevarec, H.: La valeur de la musique. Volume! **201**(1), 7–17 (2023)
39. Parasie, S., Shulz, S.: Le numérique au service de la transition écologique? Réseaux **244**(2), 11–45 (2024)
40. Peters, A.K., Capilla, R., Coroamă, V.C., Heldal, R., Lago, P., Leifler, O., Moreira, A., Fernandes, J.a.P., Penzenstadler, B., Porras, J., Venters, C.C.: Sustainability in computing education: A systematic literature review. ACM Trans. Comput. Educ. **24**(1) (Feb 2024)
41. Remy, C., Bates, O., Dix, A., Thomas, V., Hazas, M., Friday, A., Huang, E.M.: Evaluation beyond usability: Validating sustainable hci research. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. p. 1–14. CHI '18, Association for Computing Machinery, New York, NY, USA (2018)
42. Rittel, H.W., Webber, M.M.: Dilemmas in a general theory of planning. Policy sciences **4**(2), 155–169 (1973)
43. Roy, P., Gremaud, B.: La matrice interdisciplinaire d'une question scientifique socialement vive comme outil d'analyse a priori dans le processus de problématisation. Formation et pratiques d'enseignement en questions pp. 125–141 (2017)
44. Schuler, D., de Valk, M., Urvashi, S., Rose, S.: A pattern language for the limits community: We make the road by walking, a messy ethnography. In: Eighth Workshop on Computing within Limits 2022. LIMITS (2022)
45. Seurat, C., Tari, T., Beaudouin, V., Casanova, V., Cerri, L., Fournout, O., Meulemans, G., Mosgalik, I., de Mourat, R., Ricci, D., Schwartz, P.: Controverses mode d'emploi. Presses de la fondation nationale des Sciences Politiques (2021), <https://controverses.org/mode-demploi/>
46. Standl, B.: Solving everyday challenges in a computational way of thinking. In: International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. pp. 180–191. Springer (2017)
47. Tedre, M., Simon, Malmi, L.: Changing aims of computing education: A historical survey. Computer Science Education **28**(2), 158–186 (2018)
48. Tozzi, M.: Développer le jugement moral et la citoyenneté des élèves par la discussion à visée démocratique et philosophique (dvdv). Spirale-Revue de recherches en éducation **62**(2), 63–71 (2018)
49. Turner, R.: Computational Artifacts: Towards a Philosophy of Computer Science. Springer (2018)
50. Varela, F.: Autonomie et connaissance: essai sur le vivant. Seuil (1989)
51. Wegner, P.: Research paradigms in computer science. In: Proceedings of the 2nd International Conference on Software Engineering. pp. 322–330 (1976)