



HAL
open science

Dossier : "L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques"

Cédric Neumann, Camille Paloque-Bergès, Loic Petitgirard

► To cite this version:

Cédric Neumann, Camille Paloque-Bergès, Loic Petitgirard. Dossier : "L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques". Cahiers d'histoire du Cnam, vol.15 (1), 253 p., 2022, L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques. halshs-04130635v1

HAL Id: halshs-04130635

<https://shs.hal.science/halshs-04130635v1>

Submitted on 16 Jun 2023 (v1), last revised 5 Jul 2023 (v2)

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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

le **cnam**

Cahiers d'histoire du Cnam

• vol. 15

Dossier

L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques

Coordonné par Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard

2022 / Premier semestre
(nouvelle série)



Cahiers d'histoire du Cnam

• vol. 15

Dossier

**L'informatique entre à l'école :
vers une histoire de l'enseignement
des sciences et techniques informatiques**

Coordonné par Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard

2022 / Premier semestre
(nouvelle série)

Cahiers d'histoire du Cnam. Vol. 15, 2022 / 1 (nouvelle série)

Dossier « L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques », coordonné par Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard.

Les *Cahiers d'histoire du Cnam* sont une revue à comité de lecture inscrite dans le champ de l'histoire des sciences et des techniques. Elle investit des questions de sociohistoire des institutions et pratiques scientifiques et techniques, avec une vocation pluridisciplinaire (notamment histoire, sociologie, anthropologie, sciences de l'information-communication, et sciences de gestion).

La revue publie des articles de recherche évalués en double aveugle (articles longs et articles de synthèse), sous forme de dossier thématique ou en varia. Elle offre également des ressources documentaires diverses : entretiens et témoignages, encadrés informatifs, notules et enquêtes menées par des acteurs.

Un comité de lecture *ad hoc* est constitué à chaque numéro.

La liste complète des lecteurs est publiée sur la page Web de la revue :

[URL : <http://technique-societe.cnam.fr/les-cahiers-d-histoire-du-cnam-696687.kjsp>]

Fondateurs (première série, 1992)

Claudine Fontanon, André Grelon

Les 5 premiers numéros de l'ancienne série (1992-1996) sont disponibles intégralement sur le site Web du Conservatoire numérique du Cnam [URL : <http://cnum.cnam.fr>]

Direction de la publication

Bénédicte Fauvarque-Cosson, *administratrice générale du Conservatoire national des arts et métiers*

Rédacteur en chef

Loïc Petitgirard

Comité de rédaction

Andrée Bergeron, Marco Bertilorenzi, Jean-Claude Bouly, Serge Chambaud, Lise Cloître, Renaud d'Enfert, Claudine Fontanon, Virginie Fonteneau, Hélène Gispert, Irina Gouzévitch, André Grelon, Pierre Lamard, Stéphane Lefebvre, Alain Michel, Cédric Neumann, Bilel Osmane, Camille Paloque-Bergès, Loïc Petitgirard, Catherine Radtka, Laurent Rollet, Ferruccio Ricciardi, Jean-Claude Ruano-Borbalan, Marianne Thivend, Henri Zimnovitch

Comité de lecture du numéro

Marie Benedetto Meyer, Éric Bruillard, Marteen Bullynck, Hamid Chaachoua, Christophe Lécuyer, Jean-Luc Metzger, Alexandre Moatti, Francesca Musiani, Laurent Rollet, Scarlett Salman, Mathieu Triclot, Vinciane Zabban

Secrétariat de rédaction

Camille Paloque-Bergès, avec la collaboration de Bilel Osmane et d'Émeline Vaudescalet-Lecouvreur

Laboratoire HT2S-Cnam, Case 1LAB10,

2 rue Conté, 75 003 Paris

Mél : camille.paloque_berges@cnam.fr

Maquettage

Françoise Derenne, sur un gabarit original créé par la Direction de la Communication du Cnam

Impression

Service de la reprographie du Cnam

Crédits, mentions juridiques et dépôt légal :

©Cnam

ISSN 1240-2745



Illustrations photographiques :

Archives du Cnam ou tous droits réservés

Sommaire

Dossier

L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques	7
Introduction	
<i>Jalons pour une histoire de l'enseignement de l'informatique en France</i> Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard	9
Témoignage - <i>Prenons la main de la mémoire.</i> <i>Ma découverte de l'informatique (1964-1973)</i> Pierre Lescanne	29
Enquête d'acteurs - <i>Une pratique active de l'informatique à l'école</i> Gilles Blain, Gérard Nowak et Didier Vaudène	55
Fonds d'archives - <i>Archéologie d'un carton de l'Informatique pour tous</i> Jean-Noël Lafargue	89
<i>Rôle d'un logiciel dans la transposition didactique du concept d'algorithme : le cas du logiciel AlgoBox en France et des programmes du lycée entre 2009 et 2019</i> Antoine Meyer et Simon Modeste	97
Fonds d'archives - <i>Le Goupil G3, point de basculement de l'industrie de la micro-informatique en France</i> Isabelle Astic	133
Article de synthèse - <i>La micro-informatique, outil et enjeu d'innovations dans les formations du Cnam (1970-1990)</i> Loïc Petitgirard et Camille Paloque-Bergès	143
Témoignage - <i>Le Cnam, berceau de la micro-informatique.</i> <i>Entretien avec Bruno France-Lanord</i> Loïc Petitgirard	157
<i>La naissance de l'informatique à l'École centrale à la fin des années 1960 : un enseignement au service de la formation de managers</i> Cédric Neumann	183
<i>L'utilisation des tests de sélection des informaticiens des années 1960 aux années 1990 : une hiérarchisation des formations et des publics</i> Michaël Vicente	207
<i><< Si quelqu'un m'explique, je vais pas comprendre. >></i> <i>Une formation d'informatique face à la pédagogie de l'autonomie</i> Mathilde Krill	225



Dossier

L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques

Coordonné par Cédric Neumann, Camille Paloque-Bergès
et Loïc Petitgirard

Introduction

Jalons pour une histoire de l'enseignement de l'informatique en France

Cédric Neumann

HT2S, Cnam

Camille Paloque-Bergès

HT2S, Cnam

Loïc Petitgirard

HT2S, Cnam

Les systèmes éducatifs des pays développés ont massivement intégré le numérique dans leurs enseignements¹. Cette pénétration a commencé en France dans la deuxième moitié des années 1960 par « *volonté politique d'introduction puis d'intégration de l'informatique dans le système éducatif* » (Baron, 1990). L'informatique est apparue d'abord dans l'enseignement supérieur, pour les cursus d'ingénieurs et de techniciens, puis est devenue, à partir des années 1970, et surtout dans les années 1980, un enjeu

d'enseignement pour le premier et le second degré. La diffusion des ordinateurs dans les systèmes scolaires s'est opérée dans le cadre de l'allongement de la scolarité et de l'accroissement des effectifs scolaires. Elle accompagne la création du « collège unique » puis l'autonomie des établissements à partir de 1983. À cette transformation du système scolaire, s'ajoute la volonté de construire une industrie informatique française d'abord dans le cadre du Plan calcul (1966-1975). Dans les années 1980, le ministère de l'Industrie voit dans l'expansion de l'équipement informatique du système scolaire une opportunité de faire apparaître un champion national, la com-

¹ Les auteurs tiennent à remercier chaleureusement Catherine Radtka et Renaud d'Enfert pour leur relecture attentive et leurs commentaires et conseils avisés.

pagnie Thomson, dans la production de micro-ordinateurs.

Au sein d'une institution comme le Conservatoire national des arts et métiers, les « cycles d'information »² annuels sur la mécanographie³, organisés à partir de 1948 par le professeur de la chaire Techniques financières et comptables des entreprises André Brunet dans le cadre de l'Institut national des techniques économiques et comptables (INTEC), ont constitué un des socles sur lequel s'est développé un enseignement de l'informatique appliqué à la gestion. Le brevet de programmeur sur calculatrices électroniques qui est créé en 1957 par Alexis Hocquenghem, professeur sur la chaire Mathématiques appliquées, à la demande des constructeurs de matériels, est l'étape décisive vers l'intégration de l'informatique au cursus du Cnam. En effet, l'enseignement s'y fait plus théorique que les formations habituelles des constructeurs, lesquels assurent des travaux pratiques

2 Les « cycles d'information » du Cnam étaient destinés à des utilisateurs dans lesquels d'autres utilisateurs présentent leurs expériences : ils ne sont pas intégrés à un cursus d'enseignement.

3 La mécanographie désigne l'ensemble des machines permettant la rationalisation du travail administratif dans une perspective d'augmentation de la productivité. Après la Seconde Guerre mondiale, la notion tend à correspondre plus étroitement aux machines comptables et aux grands ensembles à cartes perforées. Dans les années 1950-1960, les ordinateurs sont considérés initialement par les gestionnaires comme une nouvelle série de machines mécanographiques prolongeant et amplifiant les possibilités des grands ensembles. À partir du milieu des années 1960, ils sont assimilés à un élément d'un phénomène nouveau, l'informatique, dont les propriétés managériales sont construites en opposition à la mécanographie (Neumann, 2013).

aux élèves. Le succès rencontré par le brevet conduit à la création en 1962 d'un diplôme d'études supérieures techniques (DEST), comportant deux spécialités : calcul scientifique et gestion. Cette première institutionnalisation de l'enseignement de l'informatique est achevée par la création de l'Institut d'Informatique d'Entreprise (IEE) en 1967, habilité à délivrer un titre d'ingénieur (Neumann, 2013). Les « Mathématiques appliquées » sont le berceau des formations plus spécialisées, notamment celles des ingénieurs en informatique, au sein du département qui deviendra le département Mathématiques-Informatique (Neumann, Paloque-Bergès & Petitgirard, 2016). Parallèlement, des expérimentations portent sur des dispositifs d'enseignement médiatisé par les technologies, de la télévision à l'informatique, dont le Conservatoire marquera l'histoire – avec par exemple ses écrans d'ordinateurs qui deviennent des « *super tableaux noirs* » (Baron, 1990), et le programme « Télé-Cnam », un enseignement télévisé diffusé sur l'ORTF, comprenant un cours d'informatique très suivi (Hayat & Petitgirard, 2014). Ce cheminement institutionnel n'est pas strictement illustratif de l'état des choses en France, mais est néanmoins à l'image d'une pénétration hétéroclite et diffractée de l'informatique dans le supérieur. Dans le cas du Cnam, l'informatique a remis en cause les frontières entre les enseignements techniques, mathématiques et économiques qui structuraient et hiérarchisaient les parcours de formation, et s'est rapidement mise au service de nouvelles formes et méthodes pédagogiques.

Un certain nombre de travaux socio-historiques sur les filières spécialisées d'enseignement à l'informatique dans le contexte industriel ont ouvert plus largement ce champ de recherches (notamment Grossetti, 1995 ; Neumann, 2013). La conjonction académico-industrielle est patente dans les premiers enseignements de l'informatique et assistés par l'informatique. La formation se fait alors *in situ*, ou plus exactement s'invente dans le cadre de la construction d'ordinateurs et de leur programmation directement « sur pièces ». Alors que la discipline « informatique » n'existe pas encore, c'est un recrutement *ad hoc* selon des compétences largement développées au sein de formations de haut niveau aux mathématiques⁴, donnant lieu à des moments atypiques dans l'histoire de l'enseignement de ce nouveau domaine – comme en atteste la formation et de la contribution des « *filles de l'ENIAC* », épisode illustrateur d'une conjonction encore déparée des normes qui vont réduire socio-démographiquement l'entrée dans la discipline alors que le métier se formalise et se valorise dans les décennies suivantes (Light, 1999). Ce moment original est rapidement suivi, aux États-Unis, de premiers développements de formation guidés par des constructeurs, dès 1958 chez IBM ; en France, Bull et IBM, à partir de 1957, participent

à l'enseignement du brevet de programmation du Cnam (Neumann, 2013). Puis, les expérimentations universitaires sont financées, en particulier en équipement matériel, par des firmes informatiques, comme par exemple sur le projet états-unien PLATO (*Programmed Logics for Automatic Teaching Operations*, puis *Programmed Learning and Teaching Operations*) en 1960 à l'université d'Illinois, en collaboration avec Control Data (Rankin, 2018). Ces diffusions et appropriations universitaires, notamment dans les travaux comme les loisirs étudiants, sont une première expérience de « popularisation » de l'informatique, comme l'avance l'historienne Joy Lisi Rankin, à travers l'usage de terminaux reliés à un ordinateur central qui permet de s'essayer à l'écriture de la programmation ou à la conversation électronique. En France, à partir de la deuxième moitié des années 1960, les constructeurs ainsi que les SSII demandent une prise en charge par l'enseignement supérieur de la formation professionnelle à l'informatique de manière à réduire leurs propres activités dans ce domaine (Neumann, 2013). Les comparaisons internationales entre ces pénétrations et appropriations des innovations industrielles dans les institutions du savoir (et les allers-retours et réciprocity qui en découlent) restent encore à être étudiées, dans la mesure où le panorama historique des débuts de l'enseignement de l'informatique est encore tributaire d'éclairages ponctuels sur des projets spécifiques – à l'exception des histoires états-uniennes des grands programmes de développement de sciences informatiques

⁴ Plus généralement, la voie mathématique est longtemps un canal de redirection pour l'initiation à l'informatique ; c'est un des facteurs d'explication de la difficulté pour les publics féminins à entrer dans la filière, la voie scientifique en général et mathématique en particulier étant historiquement occupée largement par les publics masculins (Bencivenga, 2012, p. 70).

comme ceux mis en place par l'Association of Computing Machinery (ACM), largement exploités par les historiens⁵. En France, l'histoire de l'enseignement de l'informatique est encore largement portée par l'association Enseignement Public & Informatique (EPI), en activité depuis un demi-siècle⁶. Fondée en 1971 par les premiers enseignants formés à l'informatique lors de l'expérience des « 58 lycées »⁷, elle dispose d'une position d'expertise dans les usages pédagogiques de l'informatique qui lui permet d'être consultée lors de la mise en œuvre des plans d'informatisation par le Ministère de l'Éducation nationale ainsi que par la Direction des lycées⁸ et celle des collèges.

⁵ Voir par exemple les publications de la revue *IEEE Annals of the History of Computing* [URL : <https://ieeexplore.ieee.org/>], organe de référence de l'histoire de l'informatique, en particulier auprès de la communauté états-unienne de sciences informatiques dont la revue est issue.

⁶ Association Enseignement Public & Informatique [URL : <https://www.epi.asso.fr>].

⁷ L'expérience des « 58 lycées » a été initiée en 1970 par la formation approfondie à l'informatique d'enseignants de toutes les disciplines. Ces derniers devaient introduire la démarche informatique dans l'enseignement de leurs disciplines notamment par la réalisation de logiciels pédagogiques. Leurs travaux sont coordonnés par l'INRP et sa section Informatique et Enseignement. Ensuite, entre 1972 et 1976, le Ministère fournit à 58 lycées des moyens spécifiques pour s'équiper en matériel informatique.

⁸ En 1974, la Direction générale des enseignements élémentaires et secondaires (DGESCO) est supprimée et est remplacée par trois entités : la Direction des écoles, la Direction des collèges et la Direction des lycées. La Direction des collèges prend en charge la pédagogie relative au premier cycle, la gestion de la vie scolaire et l'élaboration de la carte scolaire. La Direction des lycées élabore la politique d'enseignement du second

Ce numéro des *Cahiers d'histoire du Cnam* a l'ambition de poser quelques jalons dans l'évolution de l'enseignement de l'informatique, depuis son introduction jusqu'à son institutionnalisation et positionnement comme savoir légitime, discipline académique et domaine de recherche et développement (R&D) ; ainsi que d'en tirer quelques problématiques fondamentales pour des recherches futures. S'il ne traite pas de la question plus large des usages innombrables du numérique et de la numérisation généralisée des systèmes scolaires, il présentera matière à réflexion pour ce qui relève de l'informatique au service de l'enseignement (EAO, pour Enseignement assisté par ordinateur⁹). Un premier jalon consiste, à travers les pages qui suivent, à rendre visibles les recherches qui prennent pour objet les pratiques et théories de l'enseignement de l'informatique et sciences du numérique, et/ou l'inclusion de celles-ci dans des enseignements d'autres disciplines. Si le numéro ne peut, en l'état des recherches actuelles, couvrir tous les sujets, il souhaite cependant lancer quelques interrogations sur le sujet, ceci à tous les niveaux et dans une variété de formes scolaires (y compris les apprentissages formels, non formels et informels¹⁰),

cycle à la fois de l'enseignement général et de l'enseignement technologique. En 1986, la Direction des collèges et celle des lycées fusionnent pour créer la Direction des collèges et des lycées. Entre 1981 et 1985, le poste de directeur des lycées est occupé par Claude Pair, professeur d'informatique à l'Institut polytechnique national de Lorraine.

⁹ De l'anglais *Computer-based training* ou CBT.

¹⁰ CEDEFOP, *Terminology of European education and training policy. A selection of 100 key terms*. Luxembourg : Office for Official Publications of the

ainsi que dans la diversité des enseignants et publics apprenants.

Alors que l'on parle de « numérisation » croissante des sociétés, l'enseignement du numérique – ne parlons même pas de l'enseignement de ses usages ! – concerne le mille-feuille complexe des systèmes informatiques – de l'infrastructure aux applications, du matériel au logiciel, des machines aux réseaux des systèmes et réseaux numériques, et varie selon l'approche disciplinaire et le lieu et la spécialité d'enseignement. La définition de P. Dreyfus de 1962 du néologisme « informatique » (contraction d'« information » et « automatique ») recouvrait « *un ensemble de sciences et de techniques relatives au traitement automatisé de l'information* » (Legrenzi, 2015, p. 22). Une nouvelle définition consensuelle au tournant du XXI^e siècle inclut les « *processus algorithmiques qui décrivent et transforment l'information : leur théorie, l'analyse, la conception, l'efficacité, la mise en œuvre et l'application* » (Grandbastien, 2012).

Autour d'un modèle occidental de référence, regroupant des scientifiques et des industriels, porté par la société savante de l'IFIP (*International Federation for Information Processing*) – qui est parrainée par l'UNESCO depuis 1960 - la « *computer education* » (éducation à l'informatique) se diversifie rapidement dans

ses objets comme ses formes, notamment sous l'influence des questions générales d'éducation intégrant l'informatique :

La lecture des actes de la première conférence organisée par l'IFIP sur l'informatique et l'éducation à Amsterdam en 1970 (1-IFIP 70) montre déjà que les contributions sont regroupées autour de deux grands thèmes : « *computer education* » (la plus importante en volume), et « *uses of computers in education* ». Les communications à la seconde, en 1975, révèlent une grande diversité (et donc une activité multiforme), que ce soit sur la formation de spécialistes en gestion, sur l'enseignement de la programmation, sur l'« Enseignement Assisté par Ordinateur » et la « technologie éducative », ou sur des thèmes qui devaient bientôt se développer, comme « *computer literacy* » et « *computer awareness* » (Baron, 1990, p. 62).

La question de l'apprentissage des langages informatiques – que l'on parle de « programmation » ou de « code », selon le contexte – en tant que nouvelle forme d'alphabétisation, littéracie fondamentale en ce début de nouveau millénaire, est au cœur des débats sur l'école depuis maintenant plusieurs décennies (David, 2015). En effet, l'informatique comme savoir à transmettre comprend non seulement, au-delà de ses aspects logiques et mathématiques, une conception numérique de l'information, mais s'ouvre aussi à des enjeux plus généralement culturels, comme le montre par exemple la prise en charge de la thématique sous un angle éthique et culturel par l'Unesco depuis les années 1980 (Polhe, 2015).

European Communities, 2008 [URL : <http://europass.cedefop.europa.eu/fr/education-and-training-glossary>].

Le rôle de l'enseignement de l'informatique dans les transformations contemporaines des systèmes scolaires, la manière dont celles-ci ont été saisies par des politiques scolaires et la genèse de ces dernières demeure un champ de recherches encore peu exploité par les sciences sociales malgré l'omniprésence du numérique. Les questions des politiques d'incitation, d'encadrement et d'orientation de cette transmission, ainsi que celle de la réception de celles-ci, de l'appropriation des compétences informatiques par les métiers de l'enseignement (des professeurs et enseignants associés aux documentalistes et personnels administratifs), les pratiques amateurs et la diversité de situation d'apprentissage dans une variété de contextes socio-économiques et culturels sont donc posées en creux de ce numéro.

Les perspectives historiques présentées dans ce volume sont issues de diverses disciplines – portant sur une période allant des années 1950, avec l'entrée en scène des formations pré-informatiques (Neumann, 2013) jusqu'à aujourd'hui. Et comme à son habitude, la revue s'ouvre à des approches documentaires.

L'informatique entre à l'école : enseignement des sciences du numérique dans le premier et le second degré

Un premier aspect du développement du numérique dans l'enseignement renvoie à l'analyse historique et sociolo-

gique de l'introduction de l'informatique dans les *curricula* scolaires du second degré puis du premier.

En France la période, qui s'étend des années 1970 à la fin des années 1980 dans la lignée du Plan Calcul (1964-1975), est marquée par la succession de plusieurs plans de promotion des usages scolaires de l'informatique dans le secondaire : l'expérience nationale des « 58 lycées » (1970-1976), le plan des « 10 000 micros » (1978-1981), le plan d'équipement en micro-ordinateurs « grand-public » (1983), celui dit des « 100 000 microordinateurs et 100 000 enseignants formés » (1984) et enfin « Informatique pour tous » (IPT) (1985-1988). Des expérimentations poussées sur l'apprentissage ludique du code – à commencer par les célèbres tortues du langage LOGO, sont mises en place dans cette période. Dans ce dossier, Gilles Blain, Gérard Nowak, Didier Vaudène, reviennent ainsi sur les expériences LOGO qu'ils ont initiées dans le cadre d'une expérimentation pédagogique de grande ampleur, les qualifiant de « *pratique active de l'informatique à l'école* ». Le « *moment LOGO* » signale l'intérêt accru des informaticiens pour une didactique de la programmation.

Cette expérience est représentative des modalités d'introduction de l'informatique dans l'enseignement en France. En effet, la reconnaissance d'une démarche de pensée informatique définie par le raisonnement algorithmique s'accompagne d'une place marginale de l'enseignement

disciplinaire de l'informatique¹¹. Celle-ci apparaît avant tout comme un outil permettant d'enseigner autrement les autres disciplines et de faciliter leur compréhension par les élèves. Cette pratique pédagogique où l'informatique n'est pas une fin en soi mais un support pour l'apprentissage de problèmes complexes se retrouve dans d'autres pays occidentaux industrialisés, comme en témoigne par exemple cet avis au ministre de l'Éducation canadien en 1983 :

Le renforcement des facultés analytiques est lié au mode d'apprentissage « par le faire » que permet cette technologie. En tant qu'outil personnel qui permet d'enregistrer les données de l'environnement et d'en simuler les relations, l'ordinateur crée un milieu propice à l'apprentissage expérimental qui développe l'esprit d'analyse. Dans la mesure où l'enseignement dispensé par le professeur respectera l'équilibre entre l'expérimentation et les calculs par informatique, d'une part, et l'acquisition des fondements de la connaissance, d'autre part, la qualité de la réflexion scientifique s'en trouvera améliorée. L'étude de la loi de Kepler par exemple, pourra être davantage centrée sur sa portée que sur

les calculs pour démontrer que la trajectoire suit une courbe elliptique. L'ordinateur utilisé intelligemment permet de libérer la pensée du « comment » pour se concentrer sur le « pourquoi » des phénomènes et le lien entre le « comment » et le « pourquoi »¹².

La faiblesse de l'enseignement disciplinaire et technique de l'informatique dans le premier et le second degré s'explique aussi par l'importance accordée à la sensibilisation au sujet. En France, dès le début des années 1970, la Délégation à l'informatique¹³ considère que cette mission de sensibilisation constitue la priorité véritable de la formation en informatique devant celle des spécialistes. En effet, la sensibilisation, conformément aux analyses des Sociétés de services et d'ingénierie en informatique (SSII), doit permettre de former une population d'utilisateurs non spécialistes favorables à l'informatique et au dialogue avec les informaticiens. Sa mise en œuvre suppose la mobilisation de l'enseignement secondaire puis primaire, prioritairement, donc, axé sur les lycées dans la mesure où il s'agit

11 Une option informatique pour les lycées est créée en 1981 mais elle ne conduit à une épreuve du baccalauréat qu'en 1988 puis est supprimée en 1991. Les *Bulletins de l'EPI* permettent de reconstituer l'histoire de cette option. Comme nous le suggère Renaud d'Enfert, antérieurement on peut signaler la création du baccalauréat de technicien « Techniques informatiques et programmation » (section H) en 1968, dans le cadre de la réforme Fouchet du second cycle de 1965. Cette section n'a rassemblé que des effectifs limités. Durant les années 1970, entre 400 et 500 élèves sont, chaque année, diplômés d'un baccalauréat H.

12 « Informatique et télématique dans l'enseignement supérieur ». Avis au ministre de l'Éducation, avril 1983 [URL : <https://www.cse.gouv.qc.ca/wp-content/uploads/2021/05/50-0318-AV-linformatique-et-telematique-dans-lenseignement-superieur.pdf>].

13 La Délégation à l'informatique ou Délégation générale à l'informatique est un organisme interministériel créé dans le cadre du Plan Calcul. Elle a pour fonction de coordonner les politiques visant à développer l'industrie française de l'informatique. Associée avec le ministère de l'Éducation Nationale, la Délégation définit la politique nationale de formation à l'informatique. Elle disparaît en 1974 lors de la liquidation du Plan calcul par Valéry Giscard d'Estaing.

d'abord de sensibiliser de futurs cadres à l'informatique dans un contexte où l'unification du second degré n'est pas achevée ; elle est ainsi à l'origine de l'expérience fondatrice des « 58 lycées » (Neumann, 2013). Cette transformation de l'informatique en élément de « culture générale » sera amplifiée d'abord par les plans d'informatisation de l'école des années 1980 puis par l'expansion des ordinateurs personnels et des réseaux numériques dans les années 1990. Décentrés par rapport au pur contenu technique, réservés aux spécialistes, les apprentissages visent de plus en plus à familiariser l'utilisateur général à l'interface informatique, et l'enseignement de la matière se diversifie considérablement, dans le cadre général d'apprentissage des compétences numériques à l'école, visant à former de nouveaux citoyens, usagers, consommateurs des nouvelles technologies. Cela se voit bien avec la mise en place du brevet informatique et internet (B2i), passé à la fin du collège, remplacé en 2018 par le « *cadre de références des compétences numériques* » pour l'école et le collège et le lycée¹⁴. Cette évolution est congruente avec la volonté de faire de l'école un lieu d'apprentissage des compétences, des « *marketable skills* », c'est-à-dire des savoir-faire et des savoir-être utiles aux entreprises.

L'introduction de l'informatique à l'école s'est accompagnée d'un grand

nombre de recherches en sciences pédagogiques et de l'éducation, dont Jacques Perriault (1985) rappelle la généalogie, en rapport avec l'apprentissage assisté par machines, depuis la fin des années 1950 : à commencer par l'Institut pédagogique national¹⁵, lieu d'expérimentation pour la télévision scolaire et l'enseignement programmé, et qui abrite à partir de 1967 une Division des Applications Expérimentales et un Centre de Documentation sur l'Enseignement Programmé ; et à partir de 1971, un groupe d'études sur l'informatique pour la pédagogie. La diversification – et la spécification – des usages de l'informatique est en marche. Jusqu'en 1987, l'INRP (Institut national de recherche pédagogique) a la charge de la recherche, de la conception et de l'évaluation des logiciels pédagogiques.

Une série d'innovations techniques dans les années 1980, poussée par la diffusion de la micro-informatique, stimule la production de didacticiels, notamment les capacités graphiques, les supports (vidéodisques, CDRom...), des périphériques souples facilement appropriables (souris, stylos numériques...). L'idée de « package multimédia », née dans la décennie précédente, reçoit « *une nouvelle acception, où l'ordinateur joue désormais un rôle central pour acquérir, traiter et distribuer de l'information de manière individualisée, éventuellement à*

¹⁴ Document mis en ligne sur le site du Ministère de l'éducation nationale « Évaluer et certifier les compétences numériques », octobre 2022 [URL : <https://eduscol.education.fr/721/evaluer-et-certifier-les-comp-etences-numeriques>].

¹⁵ Ainsi nommé de 1956 à 1970, l'IPN se scinde par la suite en deux entités : l'Institut National de Recherche Pédagogique et l'Office Français des Techniques Modernes d'Éducation ; il est appelé aujourd'hui Institut français de l'éducation.

distance grâce à la télématique » (Baron, 1990, p. 64). Dans les pages qui suivent, Jean-Noël Lafargue s'essaie à un petit exercice d'« *archéologie d'un carton de l'Informatique pour tous* » où les composants d'un de ces « packages » font programme. La description minutieuse du contenu de ce carton, lié à un groupe scolaire normand dans les années 1980, donne à voir quelques vestiges matériels, en particulier l'ordinateur TO7/70, l'ordinateur le plus diffusé dans le cadre de l'opération. Il laisse ainsi entrevoir la possibilité d'une histoire vivante de cet épisode grâce aux logiciels en état de marche et d'approcher l'expérience vécue par les enseignants et les élèves des milliers d'établissements scolaires concernés. Dans une perspective patrimoniale, Isabelle Astic présente également dans ce dossier une autre série de matériels, les Goupil G3, achetés et diffusés dans les établissements dans le même cadre de l'IPT, en tant qu'éléments formant des nanoréseaux informatiques. Trois exemplaires de ces matériels font aujourd'hui partie d'un fond en provenance de la société SMT-Goupil, conservé au Musée des arts et métiers. Isabelle Astic retrace la constitution de ce fond, tout en rappelant l'histoire de la société SMT et la marque qu'elle a laissée dans l'histoire de l'industrie micro-informatique française.

Un dernier aperçu sur l'enseignement des sciences du numérique dans le secondaire est donné dans l'article d'Antoine Meyer et Simon Modeste, qui analysent le rôle du logiciel Algobox dans la transposition didactique du concept d'al-

gorithme. Il s'agit d'une analyse épistémologique et didactique de phénomènes curriculaires relatifs à l'enseignement de l'algorithmique au niveau du lycée, sur la période 2009-2019. Les auteurs montrent les difficultés d'ordre conceptuel et didactique qui touchent le projet incarné par Algobox.

Écoles, instituts et universités : formation à l'informatique et au numérique dans le supérieur

On peut se demander quels facteurs ont poussé diverses institutions – universités, écoles de commerce et d'ingénieurs – à développer l'enseignement de l'informatique, quels acteurs y ont promu le nouvel enseignement, sous quelle forme, comment celui-ci s'est intégré à des cursus préexistants et de quelle manière il les a modifiés. Les expériences informelles des personnels enseignants et techniciens de l'éducation avec les nouvelles machines informatiques et en réseau ne sont pas limitées à des expérimentations technologiquement avancées, mais elles se sont également appuyées sur des outils numériques en train de se fondre dans le quotidien des communications – comme cette tentative, parmi une myriade d'autres, d'enseigner un cours d'informatique théoriquement complexe (*Common Lisp for Artificial Intelligence*) à travers le médium du message électronique de liste de diffusion (Watt, 1995). De fait, les enseignants, en tant qu'utilisateurs de ces nouvelles technologies, sont force de proposition et d'expérimenta-

tion ; ils sont aussi bien imprégnés d'une culture collaborative enrichie par l'esprit communautaire des utopies numériques que par les savoirs et pratiques auxiliaires des sciences, comme la documentation (Le Deuff, 2010).

Des travaux séminaux ont été menés permettant de positionner les formations d'ingénieurs en informatique en écoles dans les logiques d'une économie territoriale, de délocalisation des compétences, de stimulation d'industries locales via la formation de ressources humaines qualifiées (Grossetti, 1991, 1995). D'autres ont été menés sur la genèse et l'évolution de la recherche en informatique, à travers la formation à la recherche de haut niveau dans une discipline aux contours encore incertains et la constitution d'enseignements en lien avec les besoins industriels (Grossetti & Mounier-Kuhn, 1995 ; Beltran & Griset, 2006). Il est important dans ce cadre de rappeler que des étudiants ingénieurs sont recrutés précocement comme doctorants, par exemple (Neumann, 2013 ; Paloque-Bergès & Petitgirard, 2017). Mais le milieu universitaire de l'enseignement de l'informatique est profondément marqué par l'interdisciplinarité, dans la lignée des courants cybernétiques (Le Roux, 2018) :

Pour des raisons tenant pour les uns aux utilisations de l'informatique dans l'enseignement programmé et pour les autres à un intérêt pour la programmation et les modèles informatiques du psychisme, des psychologues de l'apprentissage, behavioristes puis cogniticiens, se sont depuis longtemps

intéressés au sujet. Les spécialistes de l'Intelligence Artificielle, généralement rattachés à l'informatique, ont également mené depuis les origines des recherches sur l'apprentissage, aux confins de l'informatique et de ce qui est devenu les sciences cognitives (Baron, 1990, p. 60).

Se posent d'abord des questions d'équipements, dans une course à la modernité accompagnée d'une forme de standardisation des contenus comme des moyens. L'avis au ministère canadien déjà cité rappelle ainsi qu'en 1983, une université pionnière dans le numérique comme Stanford accueille des professeurs et employés administratifs qui sont équipés à domicile à plus de 25 % (ordinateur personnel ou terminal branché sur l'ordinateur de l'université) :

<i>Répondants %</i>	<i>Tâche</i>
81	Traitement de textes
81	Transmission de textes (ou autres choses) à une imprimante de haute qualité
72	Accès à des banques de documents (bibliothèques)
70	Stockage et gestion des données
64	Transmission et réception des communications
64	Utilisation des banques de données de Stanford

Pour contribuer à documenter le développement des formations à l'informatique dans différentes institutions, nous avons choisi de donner la parole à deux acteurs, dont les témoignages sont publiés dans ce dossier. Pierre Lescanne décrit, sur la période 1964-1973, sa propre découverte de l'informatique dans

le contexte nancéen, dans sa singularité, et le cheminement plus collectif qui a fait émerger la discipline informatique à l'université de Nancy. Bruno France-Lanord éclaire quant à lui l'impulsion donnée à la micro-informatique dans les activités de la chaire de Théorie et systèmes d'organisation, au Cnam, dans les années 1970. Il a été le collaborateur le plus proche de Bruno Lussato, professeur du Cnam titulaire de cette chaire à partir de 1970, qui a voulu faire de l'établissement un lieu tremplin de la micro-informatique en France, à travers un laboratoire dédié, des séminaires et des actions en direction des entreprises, en conjuguant micro-informatique et décentralisation des organisations.

Ce témoignage a également fourni l'occasion à Camille Paloque-Bergès et Loïc Petitgirard de produire un article de synthèse sur la formation et l'évolution des départements pédagogiques et laboratoires scientifiques du Cnam dans le domaine de l'informatique. Ils montrent comment la micro-informatique dans tous ses aspects est saisie au Cnam et par quels acteurs, faisant de cet outil et de ce tournant un enjeu d'innovations dans les formations. Ils donnent à voir que le passage par le domaine émergent de la micro-informatique, à partir de recherches et d'expérimentations d'enseignants-chercheurs comme Bruno Lussato et Bruno France-Lanord, sort de la salle de cours pour déployer un regard plus large sur les évolutions sociales dans leurs rapports aux organisations et aux politiques économiques.

De nouvelles analyses des formations en informatique dans le supérieur pourraient souligner les éventuels conflits se nouant autour de la construction des savoirs informatiques ; s'intéresser à la façon dont l'élaboration de *curricula* informatiques permet à ces établissements du supérieur de défendre des positions dans différents champs ou sous-champs (par exemple le champ économique, le milieu des grandes écoles etc.), non seulement aux niveaux locaux et nationaux, mais aussi dans ses dynamiques mondialisées. En effet, le développement des formations à l'informatique en France doit être repensé historiquement dans le contexte des incitations au « *rattrapage* » du « *retard français* » en matière de nouvelles technologies et d'informatique en particulier (Paloque-Bergès & Petitgirard, 2021 ; Bouchard, 2008). En ceci, l'enseignement en informatique rejoint les formations en gestion, puisque cette rhétorique du retard peut être comprise dans la relation problématique au modèle américain du management d'entreprise (Neumann, 2013 ; Mounier-Kuhn, 2010). Ainsi dans ce numéro, Cédric Neumann montre qu'à la fin des années 1960 et dans la première moitié des années 1970 l'informatique se développe à l'École centrale des Arts et Manufactures comme un enseignement de gestion destiné à maintenir le rang de celle-ci dans le champ des grandes écoles. L'informatique apparaît comme un des éléments les plus importants d'une politique de modernisation de son enseignement visant à faire des Centraliens davantage des managers que des techniciens.

Enfin, les écoles offrent un terrain à des investigations sociologiques sur les profils des apprenants comme sur les stratégies et politiques, ainsi que les enjeux économiques de leurs enseignements. En ceci, une observation des écoles privées est fondamentale. En effet, pour prendre l'exemple de la France, l'existence de formations publiques à l'informatique n'empêche pas de nombreuses créations d'écoles privées spécialisées dans l'informatique des années 1960 à nos jours. Mis à part quelques études sur la période très contemporaine, et leurs liens forts au secteur entrepreneurial (Vicente, 2017, 2018), ce secteur de l'enseignement privé demeure très mal connu. L'attitude de l'État envers ces écoles, leur légitimité, leurs relations avec l'industrie et leurs débouchés par rapport aux formations publiques restent relativement peu documentés. Michaël Vicente éclaire, dans ce numéro, un aspect important des conditions de recrutement et d'effet de hiérarchisation des écoles privées d'informatique entre 1960 et 2000, via l'analyse de leurs tests de sélection à l'entrée.

Les écoles les plus récentes, celles qui se revendiquent ouvertement de l'apprentissage du code (rejetant les dimensions académique, théorique et disciplinaire liée à la programmation), comme par exemple, l'École 42 fondée par Xavier Niel, sont caractérisées par une forte hostilité envers le système scolaire tout en recevant le soutien de l'État (Dupuy & Sarfati, 2022). L'analyse de ces écoles renseigne sur la manière dont l'État conçoit l'évolution des missions de l'enseignement et la

place des entreprises au sein de celles-ci. Mathilde Krill, dans nos pages, décortique le modèle pédagogique de l'autonomie (ou « *pédagogie active* ») dans l'école Codon¹⁶ à l'aune de sa réception par les étudiants, des pratiques effectives de l'encadrement et du succès du recrutement subséquent. Ainsi, les écoles privées peuvent être interrogées en termes de trajectoires scolaires de leurs élèves et étudiants, de rapport aux inégalités et à la méritocratie.

Des enquêtes historiques devraient à l'avenir compléter l'analyse ethnographique des pédagogies alternatives mises en œuvre par ces écoles privées. En effet, à partir des années 1960-1970, l'enseignement public, notamment l'INRP, a introduit des pratiques et des expériences inspirées des pédagogies alternatives (Viaud, 2005). Dans les années 1980, l'informatique apparaît comme un moyen de réaliser ces dernières. L'ordinateur est, en effet, présenté comme pouvant créer une relation pédagogique valorisant la créativité, l'autonomie de l'élève et sa confiance en soi par rapport aux interactions directes avec l'enseignant, alors que cette relation est encore marquée par l'autorité du jugement professoral¹⁷ ; cela reste toutefois une interrogation critique de recherche à poser et renseigner, ne serait-ce que pour aller au-delà des mythes classiques liés à cette autorité et au cadre scolaire. En

¹⁶ Le nom de l'école a été modifié pour garantir l'anonymat des enquêtés.

¹⁷ Ce point apparaît régulièrement dans la revue *Éducation et Informatique* (proche des éditeurs et du ministère de l'Éducation Nationale) ainsi que dans les *Bulletins de l'EPI*.

retracant le passage de ces préoccupations de l'enseignement public aux écoles privées, on pourrait retracer le processus plus général de captation des pédagogies alternatives par les politiques néolibérales (Mas, 2019).

Des politiques technoscientifiques aux politiques scolaires

Pour aller plus loin dans les problématiques esquissées dans les deux premiers axes, on doit s'interroger sur les politiques scolaires mises en place pour l'enseignement de l'informatique et des sciences du numérique. Modèles pédagogiques, modèles managériaux, modèles scientifiques... Quel rôle les acteurs étatiques ont-ils attribué à l'informatique, puis au numérique, dans les modèles d'enseignement et de formation ? Comment ces modèles ont-ils évolué depuis la politique d'incitation à la formation des ingénieurs en informatique des Trente Glorieuses ?

L'expansion des outils informatiques dans l'enseignement ainsi que l'intrusion croissante d'entreprises privées du secteur numérique dans celui-ci s'explique largement par des décisions de politique scolaire, et peuvent donc être appréhendées à travers les transformations globales de l'école et de ses missions depuis les années 1970. On peut avancer que l'enseignement de l'informatique a joué – et continue de le faire – un rôle crucial dans l'émergence d'une école néo-libérale orientée vers l'employabilité de ses élèves, leur intériorisation d'un esprit d'entreprise et la produc-

tion de l'innovation (Jones 2011 ; Clément, Dreux, Laval & *al.*, 2011 ; Tanguy, 2016). Cependant, à quelles questions scolaires l'introduction de l'informatique devait-elle répondre et via quelles solutions ? Répondre pleinement à ces questions impliquerait de saisir aussi les relations entre les politiques scolaires des États et les recommandations des institutions internationales (l'OCDE et l'UNESCO, pour les principales) ou de constructions supra-étatiques (l'Union Européenne, pour la période la plus récente¹⁸). Ces dernières ont théorisé le rôle de l'informatique dans les politiques scolaires et ont élaboré ou soutenu des projets d'informatique associant la dimension pédagogique à celle de l'innovation, au creux du modèle de société de la connaissance¹⁹.

¹⁸ L'Union européenne n'a pas d'agenda sur l'éducation qui relève du domaine réservé des États membres jusqu'au Traité de Lisbonne de 2008.

¹⁹ En 1970, le Centre pour la recherche et l'innovation dans l'enseignement (CERI) de l'OCDE est à l'origine de la tenue du colloque international de Sèvres sur le rôle de l'informatique dans l'éducation. Actuellement les recherches du CERI portent sur l'innovation systémique dans deux domaines : l'innovation systémique dans l'enseignement et la formation professionnels, et les ressources numériques comme innovation systémique. Voir OCDE, « Les ressources numériques d'apprentissage comme innovation systémique » [URL : <https://www.oecd.org/fr/education/ceri/lesressourcesnumeriquesdapprentissagecommeinnovationsystemique.htm>] et OCDE (2010), *Are the new millenium learners making the grade? Technology use and educational performance in PISA*, Paris : Éditions de l'OCDE. De même, l'Unesco s'est doté d'un bureau intergouvernemental pour l'informatique (IBI) et développe des recherches sur les possibilités d'application de l'informatique dans les systèmes éducatifs et sur le rôle de la coopération internationale dans celles-ci. Voir Unesco (2011), *Transforming Education: The Power of ICT Policies*, Paris : Unesco [URL : <https://unesdoc.unesco.org/ark:/48223/pf0000211842>].

Les enjeux politiques liés à sa promotion dans les institutions scolaires posent les questions de l'équipement en matériel et en logiciel informatiques, de la numérisation des pratiques et des méthodes, d'une approche algorithmique de l'éducation, ainsi que du recours croissant – et d'actualité – aux distanciels. Il existe des liens entre réformes scolaires et utilisation croissante de l'informatique qu'il s'agit encore de mieux dégager et qualifier, notamment pour comprendre quelle est la place pour les savoirs d'enseignements et formations en informatique et numérique dans cette dynamique d'évolution générale. Par exemple, en France, la multiplication des plans d'informatique pour l'école dans les années 1980 n'a pas conduit à la création d'un Capes et d'une agrégation d'informatique. Dans ces conditions, l'introduction de l'informatique dans l'enseignement ne repose pas sur une qualification professorale mais sur des compétences, au sens managérial en ce qu'elles s'opposent à la qualification, c'est-à-dire des savoir-faire et des savoir-être périssables devant être régulièrement réactualisés à travers la formation permanente. Cette situation pose des questions sur le statut du savoir informatique par rapport aux disciplines plus anciennes et sur les implications managériales de ce développement des compétences dans le travail enseignant. Parallèlement, entre les années 1980 et 2010, les écoles privées d'informatique se sont développées et institutionnalisées en France. Celles-ci sont porteuses d'une vive critique de l'enseignement public, de ses professeurs et de leur pédagogie.

Les écoles de codage, telle l'École 42, contribuent à associer étroitement l'enseignement de l'informatique à l'école néo-libérale, en niant la nécessité d'une qualification professorale et en connectant directement le contenu de l'enseignement à l'emploi (Dupuy & Sarfati, 2022). Ce numéro s'intéresse aussi à ce processus dans le cadre de politique d'innovation, de formation à « *l'entrepreneuriat numérique* » (Vicente, 2017).

Réceptions, expériences et transformations sociales de la transmission des savoirs et pratiques informatiques

Enfin, une perspective de sociologie historique (ou d'histoire sociale) des professions, s'intéresse à la manière dont l'informatique transforme le travail enseignant, le champ des formations, ainsi que les profils socio-démographiques des populations formées – à travers ses réceptions et appropriations.

Dans le même ordre d'interrogation que la mesure de l'implication des communautés d'enseignants, on peut interroger la réception de l'informatique (des matériels aux programmes) par les enseignants et leurs organisations représentatives. L'étude des relations entre l'informatique comme outil de gestion d'un système scolaire massifié et comme outil présumé de démocratisation de l'enseignement reste à faire. En particulier, devraient être abordés l'investissement syndical dans les plans d'informatisation

de l'école ainsi que la contribution de l'informatique au maintien de différences entre des établissements censés être identiques. En effet, dans les plans informatiques antérieurs au plan Informatique pour tous, l'obtention de décharges ou de crédits pour du matériel suppose la production de projets justifiant ces décharges : soit le projet d'établissement, soit un projet d'action éducative (PAE) portant spécifiquement sur l'informatique. Par-là, ces plans conduisent à une différenciation de fait de l'offre pédagogique selon les établissements. Ainsi, la coexistence, l'articulation et les tensions entre une logique de démocratisation de l'enseignement et une autre de gestion de systèmes scolaires massifiés constitue une piste encore peu exploitée pour aborder les politiques et les usages scolaires de l'informatique. Par exemple, en France, l'EPI se présente comme une organisation engagée qui voit dans la promotion de l'informatique pédagogique un outil de démocratisation de l'enseignement. Dans ce cadre, durant la première moitié des années 1980, l'association milite pour préserver la primauté des objectifs pédagogiques face aux intérêts industriels que le développement de l'informatique à l'école suscite. Ainsi, elle défend la production de logiciels pédagogiques par les enseignants et leur circulation gratuite au sein des établissements publics. Plus généralement, elle exerce une activité de médiation auprès des enseignants, d'une part en élaborant une abondante documentation technique et pédagogique, d'autre part en réclamant davantage d'heures de formation et de décharge pour les ensei-

gnants utilisant l'informatique. Cette médiation s'effectue aussi en faveur des élèves par le soutien aux clubs informatiques dépendant des foyers socio-éducatifs et assurant, en dehors des heures de classe, une sensibilisation, conformément aux priorités fixées dès le début des années 1970 en matière de formation à l'informatique. Enfin l'EPI entretient des liens étroits avec la Fédération de l'Éducation Nationale (FEN)²⁰. Cela contribue à expliquer le soutien initial de la FEN aux politiques d'informatisation de l'enseignement²¹. Une nouvelle fois, les études manquent tant sur la sociologie de ce milieu militant – les liens exacts entre les différentes organisations ainsi que les caractéristiques professionnelles de leurs membres qui ne peuvent être réduites à leurs appartenances disciplinaires – que sur la manière dont elles ont pu parvenir à concilier ou non défense du service public et utilisation de l'informatique.

Les domaines de l'informatique et du numérique, entre discipline scientifique et technique industrielle, se distinguent historiquement par ce qu'ils ont été l'objet d'investissements autodidactes par de jeunes chercheurs et techniciens (majoritairement en électronique, mathématiques, et physique) ainsi que par des

²⁰ La Fédération de l'Éducation Nationale est créée en 1948 lorsque ses adhérents décident de sortir de la CGT afin que les enseignants disposent d'une représentation syndicale autonome. En 1992, le SNES est exclu de la FEN.

²¹ Par exemple, Delapierre M., Péliisset E. & Vicard J. (1983). *Système Éducatif et Révolution Informatique*. Paris : Cahiers de la FEN.

enseignants de toutes disciplines. Ainsi, on pourra interroger le corps enseignant en tant qu'il s'investit de son propre chef dans les formations à l'informatique sous toutes ses formes, par exemple dans la production de logiciels pédagogiques. Ou au contraire, en tant qu'il peut, à certains moments ou dans certains espaces, résister à l'alphabétisation numérique, résistance sous-tendue par des perceptions négatives de l'ordinateur et la méfiance face à une potentielle « *déshumanisation de l'éducation par les nouvelles technologies* » (Bencivenga, 2012, p. 241). Enfin, il est notable que ces domaines ont également suscité bon nombre d'appropriations par les amateurs, des passionnés d'informatique aux communautés hackers et au-delà (Bratus, Shubina & Locasto, 2010). La question de l'influence (amicale, familiale, professionnelle et sociale plus généralement), ainsi que celle du mentor est un moteur d'apprentissage informel à la fois fort et très variable selon le contexte – même si des représentations culturelles et des contraintes économiques peuvent fortement peser dessus (Bencivenga, 2012, pp. 87-89). Il faut ainsi considérer les publics qui transmettent et apprennent les savoirs informatiques dans leur pluralité sociologique.

Si l'on ouvre la boîte noire des publics, l'on doit se demander comment les identités sociales des enseignants et apprenants, tous deux pouvant donc être considérés comme des publics à part entière, des domaines de l'informatique dépassent par leur complexité et leur variété la figure archétypale du jeune

geek mâle occidental issu des milieux éduqués et familiarisés aux sciences et technologies – une identité masculine investie dès la socialisation scolaire mais elle-même sujette à transformation (Peyron, 2019). On sait que l'apprentissage de l'informatique a été « masculinisé » au cours de l'histoire, tout comme son industrie (Rasmussen & Håpnes, 1991 ; Collet, 2005 ; Misa, 2011 ; Abbate, 2012 ; Schafer, 2017). L'inscription, très partielle et encore largement en cours, des femmes dans la construction d'un patrimoine de l'informatique, depuis les inventrices jusqu'aux étudiantes en passant par les industrielles et enseignantes-chercheuses, participe à une mémoire sociale de ces technologies où les femmes restent les « *oubliées du numérique* » (Collet, 2019). En France, cette marginalisation des femmes s'est d'abord appuyée sur les pratiques managériales qui ont construit une différence radicale entre l'informatique et la mécanographie dont le personnel de saisie était exclusivement féminin (dans la lignée des métiers de la dactylographie). Ce dernier, dans la première moitié des années 1960, a été systématiquement écarté de la formation sur le tas des programmeurs par les constructeurs et les utilisateurs au profit d'un personnel masculin. Cette pratique trouve un prolongement dans la politique nationale de formation à l'informatique et de promotion de ses métiers opérée dans le cadre du Plan calcul. Ainsi, dès le début des années 1970, les brochures produites par l'ONISEP pour décrire les métiers liés à l'informatique associent aux femmes les postes d'exécution

considérés comme « manuels » et ne requérant pas de compétences proprement informatiques (Neumann, 2013). De cette manière, le secteur informatique reprend des stéréotypes professionnels genrés déjà employés pour le travail à la chaîne (Lee Downs, 2002) et la rationalisation du travail de bureau (Gardey, 2001). Actuellement, les politiques d'inclusion des femmes menées par les écoles d'informatique ne parviennent pas à affaiblir les stéréotypes genrés qui structurent le monde professionnel des informaticiens. Ainsi, à l'École 42, la volonté d'augmenter le nombre de femmes par promotion est justifiée partiellement par un discours essentialiste selon lequel les femmes apporteraient des qualités spécialement féminines. Dans le même temps est opérée au sein de l'école une division sexuée des tâches dans laquelle les hommes endossent les rôles techniques et les femmes les tâches relationnelles (Dupuy & Sarfati, 2022). On pensera également aux populations en situation marginale par rapport à l'alphabétisation numérique : immigrés et séniors, personnes en difficulté économique, constituant par exemple, le terreau des formations *ad hoc* proposées par les structures associatives et/ou publiques telles que les Espaces publics numériques ou Pôle emploi en France. C'est une question qui, sur le plan des études sur l'éducation, semble encore sous-documentée car elle concerne aussi des problématiques de formats techniques relativement peu accessibles. De fait, la diffusion internationale de l'informatique (notamment personnelle) s'est structurée dans un mouvement de stan-

dardisation concernant tous les aspects du numérique, à commencer par l'écriture informatique face à la multiplicité linguistique. Les identités minoritaires ont peu ou tardivement été prises en compte dans les cadrages socio-démographiques ou plus généralement social, culturel et politique des apprentissages du numérique au niveau global (Paloque-Bergès, 2022), au profit d'une pédagogie visant l'excellence de « l'expert », devenu une figure de pouvoir dans les nouvelles économies numériques (Ensmenger, 2012). On suppose pourtant qu'il existe de nombreux développements au niveau local pour adapter les supports techniques à ces échelles éducatives ; mais ils sont peu accessibles, probablement car directement adressés à des populations ciblées, et leurs enjeux restent masqués par les normes internationales. En outre, il reste encore à étudier comment cela a pu influencer sur la conception même de l'enseignement des sciences et techniques du numérique, et ceci au cœur même des contenus d'apprentissage.

Bibliographie

Abbate J. (2012). *Recoding gender: Women's changing participation in computing*. Cambridge (Mass.) : MIT Press, 2012.

Baron G.-L. (1990). « Note de synthèse ». *Revue française de pédagogie*, vol. 92, pp. 57-77.

Beltran A. & Griset P. (2012). *Histoire d'un pionnier de l'informatique : 40 ans de recherche à l'Inria*. EDP sciences.

Beltran A. & Griset P. (2006). « Les chaotiques débuts de la recherche informatique » [en ligne]. *La revue pour l'histoire du CNRS*, 15 [URL : <http://journals.openedition.org/histoire-cnrs/50>].

Bencivenga R. (2012). « Femmes et hommes face à l'ordinateur. Histoires du développement d'une relation positive ». Thèse en Sciences de l'éducation de l'Université de Nanterre-Paris X.

Bouchard J. (2008). *Comment le retard vient aux Français : analyse d'un discours sur la recherche, l'innovation et la compétitivité 1940-1970*. Villeneuve d'Ascq : Presses Universitaires du Septentrion.

Bratus S., Shubina A. & Locasto M. E. (2010). « Teaching the principles of the hacker curriculum to undergraduates » [en ligne]. *Proceedings of the 41st ACM technical symposium on computer science education* [URL : <https://dl.acm.org/doi/abs/10.1145/1734263.1734303>].

Clément P., Dreux G., Laval C. & al. (2011). *La nouvelle école capitaliste*. Paris : La Découverte.

Collet I. (2019). *Les oubliées du numérique*. Paris : Le Passeur.

Collet I. (2005). « La masculinisation des études d'informatique : savoir, pouvoir et

genre ». Thèse en sciences de l'éducation, Université de Paris 10 Nanterre.

David J. (2015). « Literacy-Littéracie-littératie : évolution et destinée d'un concept ». *Le français aujourd'hui*, 3, pp. 9-22.

Dupuy C. & Sarfati F. (2022). *Gouverner par l'emploi. Une histoire de l'école 42*. Paris : PUF.

Ensmenger N. L. (2012). *The computer boys take over: Computers, programmers, and the politics of technical expertise*. Cambridge (Mass.) : MIT Press.

Gardey D. (2001). *La Dactylographe et l'expéditionnaire*. Paris : Belin.

Grandbastien M. (2012). « La discipline informatique et l'éducation à la maîtrise de l'information ». In D. Frau-Meigs, É. Bruillard & É. Delamotte (dir.). *E-Dossier de l'audiovisuel : l'éducation aux cultures de l'information* [URL : <http://www.ina-sup.com/ressources/dossiers-de-laudiovisuel/le-e-dossiers-de-laudiovisuel/>].

Grossetti M. (1991). « Trajectoires d'ingénieurs et territoire. L'exemple des hautes technologies à Toulouse ». *Sociétés contemporaines*, 6/1, pp. 65-80.

Grossetti M. (1995). *Science, industrie et territoire*. Toulouse : Presses Univ. du Mirail.

Grossetti M. & Mounier-Kuhn P.-E. (1995). « Les débuts de l'informatique dans les universités – un moment de la différenciation des pôles scientifiques français ». *Revue française de sociologie*, XXXVI, 2, pp. 295-324.

Hayat S. & Petitgirard L. (2014). « Télé-Cnam : enjeux politiques et dispositifs techniques d'une innovation pédagogique ». *Cahiers d'histoire du Cnam*, 1/1, pp. 127-140.

Jones K. (dir.) (2011). *L'école en Europe. Politiques néolibérales et résistances collec-*

tives. Paris : La Dispute.

Lee Downs L. (2002). *L'inégalité à la chaîne. La division sexuée du travail dans l'industrie métallurgique en France et en Angleterre*. Paris : Belin.

Le Deuff O. (2010). « Réseaux de loisirs créatifs et nouveaux modes d'apprentissage ». *Distances et savoirs*, vol. 8, n° 4, pp. 601-621.

Le Roux R. (2018). *Une histoire de la cybernétique en France (1948-1975)*. Paris : Classiques Garnier.

Legrenzi C. (2015). « Informatique, numérique et système d'information : définitions, périmètres, enjeux économiques » [en ligne]. *Vie & sciences de l'entreprise*, 200/2, pp. 49-76 [URL : <https://www.cairn.info/revue-vie-et-sciences-de-l-entreprise-2015-2-page-49.htm>].

Light J. S. (1999). « When computers were women ». *Technology and culture*, 40/3, pp. 455-483.

Mas J.-Y. (2019). « La récupération néolibérale des pédagogies alternatives ». In L. De Cock & I. Pereira. *Les pédagogies alternatives*. Marseille : Agone.

Misa T. J. (ed.) (2011). *Gender codes: Why women are leaving computing*. New-York (NY) : John Wiley & Sons.

Mounier-Kuhn P.-E. (2010). *L'informatique en France de la Seconde Guerre Mondiale au Plan Calcul : L'émergence d'une Science*. Paris : PUPS.

Neumann C. (2013). « De la mécanographie à l'informatique : les relations entre catégorisation des techniques, groupes professionnels et transformations des savoirs managériaux ». Thèse de doctorat en histoire, Université Paris 10-Nanterre, 2013.

Neumann C., Paloque-Bergès C. & Petitgirard L. (2016). « Le Cnam. Un lieu d'accueil,

de débat et d'institutionnalisation pour les sciences et techniques de l'informatique ». *TSI*, 35/4-5, pp. 584-600.

Perriault J. (1985). « Applications éducatives des machines à communiquer de l'INRP ». *Enfance*, 38.1, pp. 43-48.

Paloque-Bergès C. (2022). « Coder l'écriture plurilingue en numérique. Problèmes de gouvernance, enjeux de gouvernementalité ». *Terminal. Technologie de l'information, culture & société*, pp. 132-133.

Paloque-Bergès C. & Petitgirard L. (2021). « Jean-Jacques Salomon, critique précoce de "l'informatisation de la société" ». *Cahiers d'histoire du Cnam*, 14/1, pp. 133-161.

Paloque-Bergès C. & Petitgirard L. (2017). « L'équipe Systèmes (1975-1980) et la genèse de la recherche en informatique au Cnam ». *Cahiers d'histoire du Cnam*, 7-8/12, pp. 25-56.

Peyron D. (2019). « La construction de l'identité culturelle geek dans le monde scolaire. À la recherche d'une autre masculinité ? ». In S. Ayrat & Y. Raibaud (dir.). *Pour en finir avec la fabrique des garçons. Volume 1 : À l'école*. Maisons des sciences de l'homme d'Aquitaine/ Genre, Cultures et Sociétés, pp. 109-122.

Pohle J. (2016). « Information for All? The emergence of UNESCO's policy discourse on the information society (1990-2003) ». PhD Doctoral Thesis, WZB Berlin Social Science Center.

Rankin J. L. (2018). *À people's history of computing in the United States*. Cambridge (Mass.) : Harvard University Press.

Rasmussen B. & Håpnes T. (1991). « Excluding women from the technologies of the future? A case study of the culture of computer science ». *Futures*, 23/10, pp. 1107-1119.

Schafer V. (2017). « Femmes, genre et in-

formatique : une question historique ». *Bulletin de la société informatique de France*, 1024 (HS2 « Femmes & Informatique »), pp. 43-49.

Tanguy L. (2016). *Enseigner l'esprit d'entreprise à l'école*. Paris : La Dispute.

Viaud M.-L. (2005). *Des collèges et des lycées différents*. Paris : PUF

Vicente M. (2018). « Les dérives commerciales des organismes privés de formation à l'informatique et leur contrôle (1971-1990) ». *Éducation permanente*, 215 (2), pp. 209-220.

Vicente M. (2017). « Apprentissage du code informatique et entrepreneuriat : de la création d'entreprise à l'esprit d'entreprendre ». *Formation emploi*, 140, pp. 87-106.

Watt S. (1995). « Teaching through electronic mail » [en ligne]. *Knowledge Media Institute, Open University, Milton Keynes* (unpublished) [URL : <https://kmi.open.ac.uk/publications/pdf/kmi-95-11.pdf>].

Prenons la main de la mémoire. Ma découverte de l'informatique (1964-1973)

Pierre Lescanne

Professeur émérite à l'École normale supérieure de Lyon, LIP
(UMR 5668 CNRS ENS Lyon UCBL)

« *Les mathématiciens méprisent,
avec beaucoup d'aristocratie,
les calculateurs.* »

François Le Lionnais,
« Un Certain disparate »¹.

Cet article présente mon initiation à l'informatique durant la période 1964-1973 et je vais tenter de retracer en quoi consistait cette initiation, sachant que comme tous les gens de ma génération, je suis essentiellement autodidacte ; en effet, l'enseignement de l'informatique que nous recevons

avant et après 1970 est minimal. Nous prenons quelques cours certes, mais l'essentiel de notre apprentissage se fait par la lecture de livres ou d'articles, peu nombreux, parfois par la redécouverte de concepts, par quelques rares conférences et par les écoles de printemps et d'été qui se mettent en place. Notre méconnaissance de l'anglais est un énorme handicap. En effet, dans ma formation du collège et du lycée, je n'ai appris que l'allemand, le latin et le grec, ce qui n'est d'aucune aide, compte tenu du fait que, dès cette époque, même les Allemands et les Suisses publient en anglais. De plus, immédiatement acquis, les concepts et les contenus scientifiques doivent être enseignés aux autres dans cette construction ininterrompue d'une science nouvelle à laquelle nous participons.

¹ Entretien avec Jean-Marc Lévy-Leblond et Jean-Baptiste Grasset (chap. 37, 1976), publié sur oulipo.net [URL : <https://oulipo.net/fr/un-certain-disparate>].

1957	Installation d'une IBM 604, machine à programmes câblés
1958	Installation d'un IBM 650, premier ordinateur
1958	Création du cours d'Analyse et calcul numérique
1959	Création du centre de calcul
1964	Mon entrée à la Faculté des Sciences de Nancy
1965	Installation d'une CAE 510
1967	Création du département informatique de l'IUT
1968	Ma nomination comme assistant délégué (non titulaire) – mon premier cours de programmation
1970	Installation du CII-10070
1971	Ma thèse de 3 ^e cycle
1971-72	Mon service militaire
1972	Ma nomination comme maître assistant à l'Université de Nancy II
1973	Création de l'équipe de recherche associée au CNRS
1974	Mon détachement au CNRS
1976	Création du CRIN, laboratoire associé au CNRS
1979	Ma thèse d'État
1980	Mon départ au MIT

Table 1 - Les étapes de l'informatique universitaire à Nancy et de ma carrière

La table 1 retrace les principales étapes de l'émergence de l'informatique à Nancy, ainsi que ma propre carrière, qui s'y insère. Cependant, j'adopterai ici dans ma démarche une approche parfois un peu technique et globalement subjective. En effet, je n'analyserai pas l'apparition de l'informatique dans le milieu universitaire nancéien, mais je dirai comment

j'ai vécu ces années d'apprentissage et de construction de la science qu'est maintenant l'informatique. Je présenterai comment cette activité s'est, chez moi, articulée avec un enseignement de mathématiques, compte tenu de ma formation initiale et de la rareté des postes en informatique à cette époque et compte tenu aussi de mon détachement au CNRS.

Mon but est de faire connaître aux générations plus jeunes, qui ont toujours baigné dans l'informatique, ce que nous, les pionniers, avons vécu. Mais comme je ne considère que le petit bout de la lorgnette, j'invite donc le lecteur qui souhaite un peu plus d'objectivité sur les débuts de l'informatique à Nancy à écouter la conférence enregistrée de Marion Créhange² ou à lire, les écrits de Claude Pair³, de Marion Créhange et Marie-Christine Haton (2014), de Pierre-Éric Mounier-Kuhn (2010) ainsi que ceux dont je suis l'auteur ou le co-auteur (2021 ; 1996). De plus une table donne en annexe C, une description succincte des langages de programmation que j'ai appris à utiliser, tandis que l'annexe A est un glossaire des termes techniques que j'utilise.

Mes études supérieures

Je suis né à Dakar le 22 mars 1947 et j'ai fait mes études secondaires à Colmar. Au moment des études supérieures, à l'automne 1964, je choisis Nancy, parce que ma grand-mère y habite et que je peux loger chez elle, mais aussi parce que je sais qu'en ma-

thématiques c'est un endroit où s'y font de bonnes choses. Ayant lu en terminal, le livre édité par François Le Lionnais, *Les grands courants de la pensée mathématique* (1948), je sais que Nancy jouit d'une réputation d'excellence héritée du temps où elle hébergeait les bourbakistes, et je connais un peu les enjeux de la recherche en ce domaine, sans toutefois, bien sûr, les maîtriser. Quand les cours de Claude Georges, notre professeur en première année, nous conduisent à aborder les « familles sommables » (les « séries »⁴ étant considérées par lui comme un peu trop contraintes) dès le mois de décembre et un peu plus tard au mois de janvier « l'intégrale de Stieltjes » sur les « fonctions à variation bornée » (là encore parce que « l'intégrale de Riemann » sur les « fonctions continues » n'est pas assez générale), je suis moins submergé que mes condisciples, pour lesquels cette « pédagogie » de choc fait l'effet d'une douche glacée⁵.

² Marion Créhange, « Le compilateur Algol 60 sur IBM 1620 », Colloque Claude Pair, 14 juin 2019 [URL : <https://videos.univ-lorraine.fr/index.php?act=view&id=7762>].

³ Claude Pair, « Notions sur la théorie des langages » (cours rédigé par Alain Quéré), Université de Nancy, Faculté des Sciences, Institut universitaire de Calcul automatique. Archives personnelles de l'auteur.

⁴ En mathématiques, les « séries » sont les sommes infinies qui peuvent diverger ou converger. Dans les « familles sommables », il n'y a pas d'ordre sur les termes. Si l'on compare l'enseignement de Mathématique générale et physique, ou MGP, que l'on appelle propédeutique et qui est la première année de faculté des sciences, avec celui des classes préparatoires aux grandes écoles, on note que les séries n'étaient pas au programme de la première année de math-sup, mais seulement à celui de la deuxième année de math-spé, tandis que les familles sommables n'étaient pas évoquées du tout.

⁵ Je découvre plus tard que Claude George fonde son cours d'analyse de MGP, sur le livre de Walter Rudin *Principles of Mathematical Analysis* ; ce livre est destiné aux étudiants de quatrième année de bachelor ou de première de master des universités américaines. Le cours de Math I de Pierre Eymard (deuxième année de fac pour nous) est fondé sur le livre de Walter Rudin *Real and Complex Analysis* que nous nous procurons

Nous étions dans l'enseignement supérieur depuis un peu plus de deux mois et n'avions jamais entendu parler au lycée de séries et d'intégrales. En 1967, je me suis retrouvé « licencié », la licence étant le diplôme qui permettait de se présenter aux concours de l'enseignement et au DEA⁶. Mais je sais que je veux faire de la recherche et un DEA de math me paraît la chose la plus normale. Je m'y inscris par conséquent. Mais j'ai plusieurs activités annexes : je fais de la voile, je joue au rugby (avec mes 78 kg, comme je suis grand, je suis deuxième ligne !), je suis investi à l'UNEF (Union Nationale des Étudiants de France) Quand Mai 68 éclate, je suis au front à la Faculté des sciences de Nancy, où rétrospectivement on peut dire que ça a été relativement calme, quoique j'apprendrai ultérieurement que deux éminents professeurs en ont été très affectés. Nous, les étudiants syndicalistes, « reconstruisons » l'université et rencontrons les professeurs syndiqués, dont l'un d'eux est au SGEN⁷ et avec lequel, je sympathise ; c'est Jean-Louis Ovaert⁸.

d'ailleurs dans un achat groupé sur la recommandation de notre professeur. Ce livre est très clairement, dans l'esprit de son auteur, un cours de master (Bac+5). C'est d'ailleurs, dans ce livre que je me suis initié à l'anglais.

6 Le DEA était le Diplôme d'Études Approfondies. Il correspondait *grosso modo* au master d'aujourd'hui après quatre ans d'études.

7 Syndicat général de l'Éducation nationale, CFDT (Confédération française démocratique du travail).

8 René Cori, Anne Michel-Pajus et Robert Rolland, *Jean-Louis Ovaert – Un homme d'action et de convictions*. Brochure IREM-APMEP, juillet 2015 [URL : <http://numerisation.irem.univ-mrs.fr/PS/IPS15006/IPS15006.pdf>].

Rebelle⁹, je suis dans mon engagement au syndicat étudiant, et rebelle je suis en mathématiques. De jeunes assistants montent, dès 1967, un séminaire de recherche alternatif, dans une annexe du département de Mathématiques, rue Sellier à Nancy¹⁰ et nous y sommes conviés ; nous nous y rendons, Jean-Luc Rémy¹¹ et moi, comme des conspirateurs. Quand il s'agit de choisir mon sujet de mémoire de DEA, Pierre Eymard, qui le dirige, nous propose de lire un article de mathématiques, parmi une liste qu'il nous soumet, pour en faire ensuite un compte rendu écrit. Comme au séminaire alternatif, j'ai entendu parler des « mesures idempotentes » je choisis l'article de P. J. Cohen¹² de 1960, « On a conjecture of Littlewood and idempotent measures ». Mes collègues du séminaire alternatif me conseillent en fait de lire et de rapporter sur un autre article intitulé

9 Ce qualificatif m'a été donné, des années après, par Françoise Bellegarde, pour qui, je suis scientifiquement et universitairement rebelle. Françoise Bellegarde (1942-2016), qui a d'abord été professeure de mathématique en lycée, fera, avec Claude Pair, une thèse de 3^e cycle sur la compilation, puis, avec moi, une thèse d'État sur la programmation fonctionnelle. Elle séjournera plusieurs années aux États-Unis, puis sera professeure à Besançon, jusqu'à sa retraite.

10 Le département de mathématiques est situé rue de la Craffe, au cœur de la faculté des sciences où nous avons nos cours. Le bâtiment est aujourd'hui occupé par un collège.

11 Jean-Luc Rémy est le plus brillant étudiant en mathématique de ma génération. Il fera une carrière de chercheur en informatique au CNRS et sera connu pour son algorithme de génération aléatoire d'arbres binaires. Cf. Wikipédia, francophone « Algorithme de Rémy » [URL : https://fr.wikipedia.org/w/index.php?title=Algorithme_de_R%C3%A9my&oldid=182172482].

12 Médaille Fields, en 1966, mais je ne fais pas le lien.

« A simple proof of the theorem of P. J. Cohen »¹³ (Amemiya & Ito, 1964). Cela plaît à Pierre Eymard, spécialiste d'analyse harmonique, thème de l'article et donc au jury, puisque j'obtiens mon DEA sans problème en septembre 1968.

Suite au décès d'une enseignante chercheuse, dans un accident de voiture¹⁴, un poste d'assistant se libère. Claude Pair, que je ne connais pas encore à l'époque, en parle à son grand ami Jean-Louis Ovaert et je me retrouve embauché, pour enseigner les maths. Ce sera en PC1 (première année du cycle Physique-Chimie). À l'époque, les recrutements se font à la bonne franquette. Je fais un choix, pour l'année 1968-1969 : je commence à enseigner tout en préparant l'agrégation de maths, si ça passe, tant mieux, sinon, je commencerai une thèse de troisième cycle dans une discipline à définir entre maths et la toute nouvelle informatique. Ovaert m'a dit du bien de son ami Claude Pair, et les mathématiques pures, qui m'avaient beaucoup plu en tant que matière d'enseignement, me tentent moins comme domaine de recherche, car elles me paraissent ne traiter que de sujets avec un moindre intérêt social, comme c'est le cas des « mesures idempotentes ». On ne peut pas raconter à son frère ou à sa sœur ce qu'on fait et parfois même pas à un autre mathématicien. D'autre part, la recherche en mathématiques pures,

malgré la dynamique du séminaire alternatif ne semble pas être faite en équipe et, si équipes il y a, elles sont à Paris. La présence d'un vrai leader qui s'implique, d'une ambiance, d'un domaine plein de promesses me titille. Quand l'étape de l'agrégation est franchie avec succès, mon choix est fait, je ferai de l'informatique¹⁵, mais je garderai toujours de bons contacts avec les mathématiciens et avec les mathématiques.

Je suis assistant à la Faculté des sciences jusqu'en 1971, année de mon service militaire. Claude Pair enseigne le cours d'algèbre et je suis une année son assistant pour les travaux dirigés. En 1970, avec Robert Mainard¹⁶, mon professeur de physique de MGP, qui en est à l'origine, je participe à la mise sur pied d'une filière de premier cycle Mathématiques, Physique et Technologies, qui se veut être une alternative aux autres filières scientifiques, à savoir MP (Mathématiques et Physique) et PC (Physique et Chimie) par son contenu et son orientation résolument technologique. Je suis chargé d'un enseignement intégré (cours et TD) de mathématiques devant une promotion d'une vingtaine d'étudiants. J'ai 23 ans, je ne suis guère plus âgé qu'eux.

¹⁵ Nous employions le terme depuis au moins 1967, date de l'inauguration du département Informatique de l'IUT qui doit sa création au Ministre Christian Fouché, qui avait fait l'annonce d'un département de « carrière de l'informatique », au lieu de « carrière de l'information ». Les deux départements seront alors créés.

¹⁶ Robert Mainard fut ensuite le troisième président de l'Université scientifique de Nancy.

¹³ On notera le terme « *the theorem* », comme s'il n'y a aucune ambiguïté sur le théorème de P. J. Cohen dont il s'agit.

¹⁴ Joëlle Rousseau.

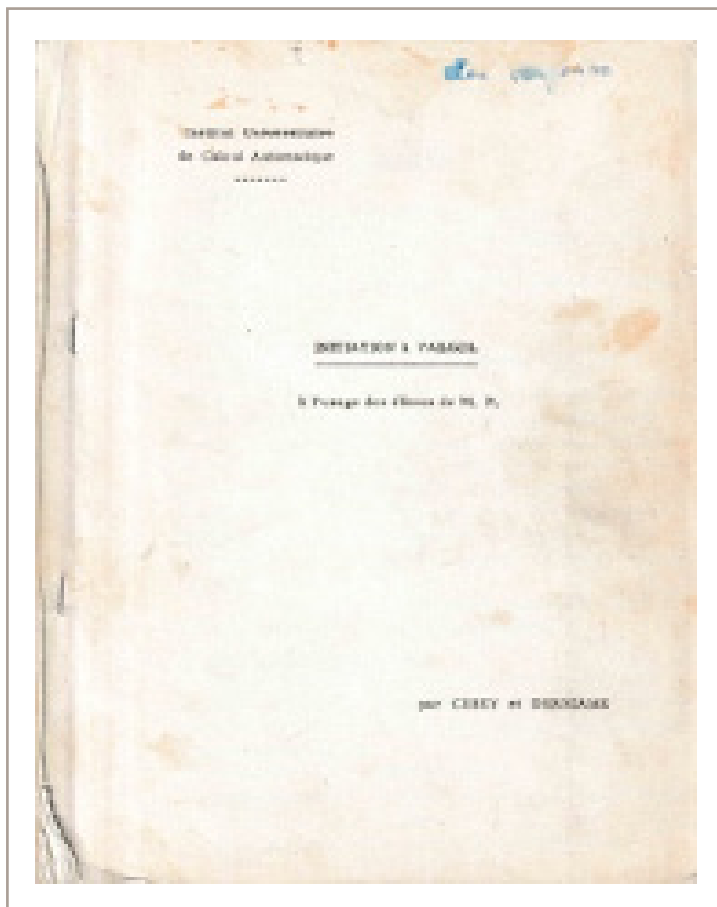


Figure 1 - Mon polycopié d'Algol 60

Mes premiers cours d'informatique

Si je veux un jour faire de l'informatique, il faut que j'apprenne la programmation ! Or, à l'automne 1968, un cours d'Algol 60 est proposé et est enseigné par Christiane Legras¹⁷. Je me souviens de ce

cours au tableau noir dans une salle du bâtiment du Centre de calcul, en face du Trésor de la langue française, avenue de la Libération, près de la place Godefroy de Bouillon. Le bâtiment est moderne,

à Nancy. Jean Legras était à l'origine mécanicien avec une thèse sur l'aile portante, mais il deviendra un spécialiste d'analyse numérique, qu'il enseigne à la faculté des sciences et dans les écoles d'ingénieur. Il est l'auteur de plusieurs livres d'enseignement de l'analyse numérique (Legras, 1963, 1968).

¹⁷ Fille de Jean Legras (Legras, 2008), fondateur du centre de calcul de Nancy et initiateur du calcul numérique

différent de ceux de l'Institut de mathématiques et de physique de la Porte de la Craffe. Nous avons eu comme support de cours le polycopié « Initiation à l'Algol » à l'usage des élèves de Maths-Physique par Cusey et Derniame¹⁸ (cf. figure 1 et plan du cours en Annexe B). Au milieu des instructions¹⁹ *début, fin, si, alors, sinon*, et des ; de cette version francisée d'Algol 60, Christiane nous propose un exemple que je ne comprends pas ; en effet, je n'ai pas fait d'analyse numérique, ni *a fortiori* de programmation, ni encore moins d'algorithmique et je ne reconnais pas, dans l'instruction $x := (x/2) + (1/x)$; la suite... qui calcule $\sqrt{2}$. Pour quelqu'un qui, comme moi, a appris une certaine dose de maths, c'est vexant ! Mais ça me fait comprendre sur quoi, en pédagogie, peut reposer un blocage. Ça me fait aussi comprendre que l'enseignement de mathématiques pures que j'ai suivi est particulièrement abstrait. Mais je me souviendrai de cette suite, car je la prendrai, comme exemple, neuf mois plus tard, pour introduire ma leçon d'agrégation intitulé suite de Cauchy ; en effet, cette suite est effectivement un bel exemple de suite de rationnels qui ne converge pas vers un rationnel. Pour calculer cette suite il faut utiliser une boucle *POUR* dont la sémantique (ce qu'elle signifie) nous est expliquée dans l'organigramme de la figure 2. Étant donné qu'aujourd'hui, j'éprouve des difficultés

à comprendre cet organigramme, je me demande ce qu'il en était en 1969. En fait, dans ce cours d'Algol 60, nous apprenons plus la syntaxe que la sémantique. Mais nous apprenons surtout comment écrire un programme qui sera accepté par le compilateur. Nous ne savons pas encore qu'affirmer qu'un programme est correct, ce n'est pas cela. Nous commencerons à le comprendre quand nous élaborerons la « Théorie des programmes ». De plus, maîtriser Algol 60, c'est maîtriser les entrées-sorties, comme cela est décrit dans le chapitre IV du polycopié (Annexe B). C'est indispensable, mais un peu rébarbatif et cela n'a pas l'élégance des autres constructions. Leur nécessité s'impose si l'on veut écrire des programmes qui produisent des résultats qui sortiront sur une imprimante, car c'est la seule sortie possible, lisible par le commun des mortels. Les autres médias, à savoir les rubans perforés et les cartes perforées ne sont pas d'une grande utilité pour tester si notre exercice a été concluant. Comme le dit le polycopié, ces instructions sont spécifiques à la CAE 510²⁰. À part cela, Algol 60 me plaît beaucoup et l'algorithmique qu'il sous-tend aussi. Avec ce cours, il n'y a pas de passage sur machine. Plus tard, j'essaie mon premier programme Algol 60 (très probablement

¹⁸ Michel Cusey et Jean-Claude Derniame. « Initiation à l'Algol à l'usage des élèves de M. P. », Institut Universitaire de Calcul Automatique, 1968. Polycopié.

¹⁹ Voir le glossaire en fin d'article.

²⁰ La CAE 510 est une machine qui n'a pas de système d'exploitation. Elle est donc mono-utilisateur et nécessite pas mal de manipulations pour la faire fonctionner comme l'on veut. Les concepteurs du compilateur ALGOL 60 de cette machine n'ont pas pu s'appuyer sur un système d'exploitation pour leurs entrées et sorties et donc ils ont dû tout concevoir à partir de zéro. De plus, le langage machine est spécifique. Ils n'ont pas pu envisager une réutilisation de leur travail.

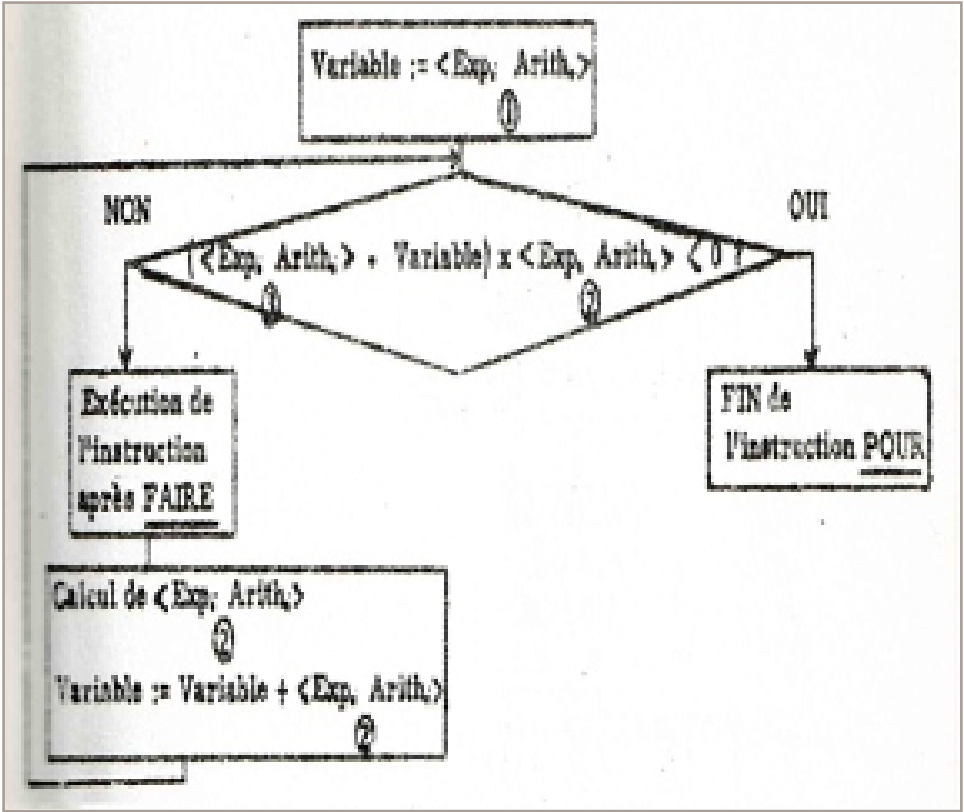


Figure 2 - L'explication de la boucle POUR par un organigramme, dans le polycopié

un calcul de factoriel) sur la CAE 510 du centre de calcul avec sortie sur son imprimante, qui est une machine à écrire IBM à boule (figure 7). Auparavant, il aura fallu charger le compilateur qui est sur un ruban de papier perforé. Tout un art ! Il ne faut surtout pas le déchirer ! Quelque temps après, je suis un cours de COBOL, mais je comprends rapidement que ce langage n'est pas pour moi. Je n'en retiens que ses quatre divisions et ses PIC 999, car, si j'ai bien compris, l'important dans ce langage, ce sont les formats.

Mon ami Jacques Guyard²¹ m'a raconté une anecdote qui illustre bien le monde informatique d'alors. Fasciné par un cours d'Algol 60 (peut-être le même que le mien) où on avait dû lui exposer un autre grand classique du genre, à savoir le triangle de Pascal, il veut passer à la pratique. Or il vit dans un monde de physiciens où le langage de référence est

²¹ Par la suite, Jacques Guyard est devenu enseignant-chercheur en informatique, professeur et directeur de l'École d'ingénieur ESIAL, qui deviendra Télécom Nancy.

FORTRAN. Son laboratoire possède des programmes qui calculent des nombres similaires utilisés par les physiciens. Ces programmes et leurs sous-programmes ne sont pas récursifs et Jacques Guyard s'est dit qu'en les rendant récursifs il en simplifierait le code. C'est ce qu'il fait. Le compilateur ne réagit pas à ces appels imbriqués de sous-programme et produit un code exécutable. Mais bien sûr, ce code fait n'importe quoi. En se renseignant, il comprend que FORTRAN à la différence d'Algol 60 n'autorise pas les appels imbriqués de sous-programme, ce qui est le principe de la récursivité et que le compilateur FORTRAN ne prend pas la peine d'en avertir l'utilisateur, tant la récursivité paraît incongrue, mais il « compile » néanmoins le programme. En FORTRAN, un appel à un sous-programme n'est rien de plus qu'un branchement²².

Pour compléter ma formation, je prends un « vrai » cours de programmation : le certificat C1 de la maîtrise d'Informatique qui vient d'être créée. J'y apprend les fondements de la programmation de l'époque à savoir les registres, l'adressage indirect, le calcul

d'instruction, les instructions de base comme LOAD ou les branchements, bref le langage machine ainsi que l'assembleur et son langage. J'y apprend aussi ce qu'est un système d'exploitation ou comment une machine « boote ». Le professeur est Jean-Claude Derniame, maître-assistant²³ à l'époque et son assistant pour les travaux dirigés est Jacques Ducloy²⁴, qui a étudié à l'ENSEM quand j'étais à la faculté. Il a suivi un cours similaire, l'année précédente, ce qui l'autorise à être notre enseignant. D'une année sur l'autre le contenu du cours évolue au gré de l'acquisition des connaissances et des progrès de la technologie. Donc on ne peut même pas dire que Jacques ait suivi le même cours l'année précédente, mais il prend sa fonction très à cœur. Ainsi comme le centre de calcul a acquis un nouvel ordinateur : un CII 10070, nous apprendrons son langage d'assembleur, le SYMBOL, et son métalangage d'assembleur, le METASYMBOL. La documentation de ces langages, surtout du METASYMBOL est insuffisante et probablement incohérente, si bien que nous devons nous habituer à voir la doxa changer d'une semaine sur l'autre. Nous coopérons avec notre enseignant et ami pour essayer de dégager les concepts de ce langage qui mélange,

²² Le grand spécialiste de la récursivité, Henry Gordon Rice (le célèbre théorème de Rice est une de ses contributions majeures), écrit en 1965 un article (Rice, 1965) où il montre, ce dont on n'est pas certain à l'époque, à savoir qu'en FORTRAN on peut coder n'importe quelle fonction récursive au sens mathématique du terme, au prix du codage de la pile de récursion par des tableaux. Il implante pour cela la célèbre fonction d'Ackermann, dont il donne un programme FORTRAN d'une trentaine de lignes et un organigramme pour en décrire la structure, alors que le code Algol 60 de la même fonction tient en quelques lignes et n'a pas besoin d'explications supplémentaires.

²³ Il devenu professeur à l'Université de Lorraine.

²⁴ Jacques Ducloy a fait l'*École nationale supérieure d'électricité et de mécanique* (ENSEM). Il deviendra ingénieur de recherche au CNRS, d'abord à l'IUCA (Institut universitaire de calcul automatique), puis à l'INIST (Institut de l'information scientifique et technique), puis au LORIA, qui est le laboratoire de recherche en informatique de l'Université de Nancy.

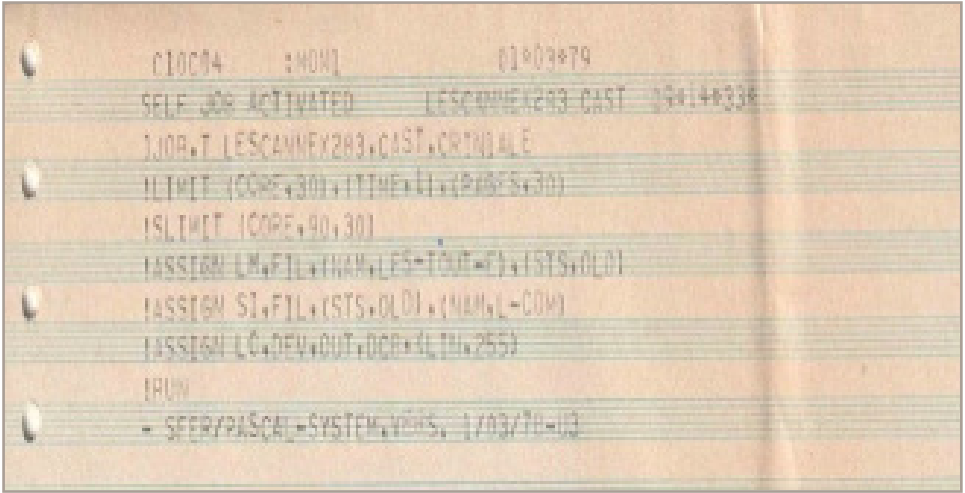


Figure 3 - Listing avec ses « cartes de commandes »

dans la même syntaxe approximative, le langage d'assembleur et ce que l'on peut dire sur lui. En effet, puisqu'en assembleur, on répète plusieurs fois les mêmes instructions à des nuances près, une suite d'instructions en METASYMBOL, n'est pas un programme, à proprement parler, mais une description visant à produire, puis à faire interpréter un programme d'assembleur en SYMBOL. On s'y perd un peu, entre tel mot clé qui appartient au langage d'assembleur et tel autre qui appartient à son métalangage. Qu'est-ce que dit le manuel ? Le dit-il vraiment ? D'ailleurs nous n'avons que des photocopies d'extraits parmi ceux qui sont pertinents. Et puis, ce cours n'est que théorique, puisque nous n'avons pas encore la machine.

Nous apprenons aussi les cartes de commande du système d'exploitation, qui sont des consignes à l'ordinateur.

On parle de « cartes » puisque *a priori* nous entrons nos programmes par cartes perforées. Même quand nous rentrons les programmes à la console, la syntaxe bizarre par carte (ou ligne) de commande sera conservée. La figure 3 est issue d'un listing relatant, plusieurs années plus tard, l'exécution du programme Pascal de calcul symbolique dont je parle plus loin. Puisque le système d'exploitation fonctionne en traitement par lot²⁵, il faut lui dire comment prendre en compte mon travail (mon *job*) quand ce sera le moment, pour lui, de le faire. La syntaxe des « cartes » est incohérente et incompréhensible et n'a absolument aucune logique. La seule cohérence entre toutes les cartes est le fait qu'elles commencent par un point d'exclamation « ! », suivi du nom de la commande. Il y a les cartes

²⁵ Je découvrirai les systèmes en temps partagé au Massachusetts Institute of Technology (MIT) en 1980.

LIMIT et SLIMIT qui énoncent les limites en temps, en mémoire interne et en mémoire externe dans lequel mon programme a le droit de s'exécuter et les tailles des sorties qu'il a le droit d'imprimer²⁶. Si je choisis des limites trop grandes, mon programme sera rejeté du lot. Il y a la carte RUN qui demande l'exécution d'un programme donné, ici le sous-système SFER/PASCAL-SYSTEM qui sert d'environnement à mon programme. La pire des cartes de commande est la carte ASSIGN qui indique où lire les données, soit sur un disque fixe, soit sur un disque amovible, soit sur un lecteur de bande magnétique, soit sur un lecteur de cartes perforées et où produire les sorties sur les mêmes, auxquels il faut ajouter une imprimante à tambour. Chaque support physique (*device*) a sa propre syntaxe, puisque, avec le recul, j'ai l'impression qu'elle a été conçue sur un canevas commun par des programmeurs différents qui l'ont définie au moment d'écrire cette partie de logiciel. Mon *listing* de la figure 3 comporte trois cartes ASSIGN. Si ma mémoire est bonne, l'une dit où le système doit trouver mon programme à exécuter, c'est-à-dire dans le fichier LES-TOUT-E, où il doit trouver les données de mon programme, où il doit écrire les résultats de mon programme, en l'occurrence sur l'imprimante à tambour et que ladite imprimante utilise du DCB, c'est-à-dire du « décimal-codé-bi-

naire »²⁷. La figure 3 illustre bien le type de sortie produit par une telle imprimante. Quand le programme est rejeté du lot à cause d'une carte de commande erronée, c'est l'humiliation, car il faudra attendre le prochain tour (le prochain lot), qui peut être dans un quart d'heure ou une demi-heure ou le lendemain. Nous sommes très conservateurs et nous nous contentons de recopier les exemples qu'on nous a donnés sur des pages manuscrites photocopiées, en n'essayant de ne commettre aucune erreur dans les parenthèses, les virgules et les mots clés et en essayant d'identifier les parties qui changent pour notre programme et celles qui ne sont pas susceptibles de changement. Il est clair que cette pédagogie n'est pas basée sur des principes et des concepts, mais sur des exemples à imiter, un peu comme les scribes babyloniens apprenaient l'algorithmique (Knuth, 1972).

Pour l'initiation pratique, nous avons, comme pour toute formation technologique, ce que nous appellerions aujourd'hui un projet semestriel, en l'occurrence un projet logiciel²⁸ puisqu'il s'agit d'une formation à l'informatique. Nous devons écrire un assembleur (un programme de traduction) d'un langage d'assembleur *ad hoc* vers une machine

26 Il n'est pas rare malgré ces contraintes de voir l'imprimante échapper à son dresseur et produire en musique des pages de 0 et de 1. Cette musique anticipe celle des *Tambours du Bronx*. Ça fera du papier pour les classes de maternelles !

27 On notera que les restes de la carte utilisent l'anglais, mais que l'abréviation DCB est du français, puisque « décimal codé binaire » se dit en anglais *binary coded decimal* et s'abrège en BCD.

28 Le mot « logiciel » venait d'être créé, nous ne l'utilisons pas encore. Il aura plus de succès que les noms « binon » (*bit*) et « bogue » (*bug*) créés en même temps.

virtuelle²⁹ et donc simulée dite machine C. Le tout est écrit dans un autre langage d'assembleur. C'est un processus un peu fastidieux, car le langage n'est pas convivial, mais réaliste, quoique, dans un but didactique, il soit nettement simplifié par ses concepteurs, par rapport à un langage d'assembleur réel de l'époque. Il n'en demeure pas moins que cette activité m'a beaucoup appris. Avant l'arrivée du CII 10070, nous avons la possibilité de passages sur la CAE 510, la seule machine qui nous est accessible. Pour cela, il faut écrire le programme sur des bordereaux *ad hoc*, puis transmettre ces bordereaux à une perceuse³⁰ qui nous rend dans un casier un paquet de cartes ; puis munis de notre paquet de cartes, nous pouvons accéder à la CAE 510, dans une salle spéciale qui lui est dédiée. Auparavant, il faut réserver la salle. Mon groupe de projet est composé de Roger Mohr³¹ et de Marcel Navet³². Le seul créneau disponible, que nous voulons suffisamment long pour pouvoir faire plusieurs passages en cas d'erreurs, est la nuit du

samedi au dimanche. L'un d'entre nous a retiré, auprès de la secrétaire, la clé du centre de calcul qui nous permettra de rentrer et sortir nuitamment. Nous commençons le long processus : d'abord en démarrant la CAE 510 en positionnant les *switchs* du mot sur lequel l'ordinateur *boote*. Le programme qui démarre l'ordinateur est sur une bande magnétique, donc nous n'avons rien à faire qu'attendre que la phase de démarrage soit complète. D'ailleurs « attendre » est l'une de nos principales activités. Pour le compilateur du langage dans lequel la machine C est simulée, il nous faut le charger, car il est sur un ruban perforé. Le lancement de la lecture se fait sur la machine à écrire IBM à boule qui est la console de contrôle opérateur. Les différentes phases du processus sont décrites dans un classeur qui doit rester toujours près de l'ordinateur, sur une table. La plupart du texte est dactylographié, mais des modifications ont été ajoutées au stylo à bille après qu'ont été raturées les consignes obsolètes. Nouvelle attente ! Puis sur une nouvelle instruction à la console, la CAE engloutit notre paquet de cartes perforées. Nouvelle attente ! La machine simulée commence à lire notre programme (nos données), puis se met à calculer. Chaque phase de calcul a son propre bruit. Avant la lecture de la bande magnétique c'est relativement silencieux, seul le clignotement des diodes rouges montre que l'ordinateur calcule. Quand vient son tour, on entend surtout le léger son du dispositif qui la maintient tendue, en gros un petit *tchak* toutes les demi-secondes. La lecture du ruban

29 « Virtuelle » signifie qu'il ne s'agit pas d'un vrai ordinateur mais d'un ordinateur fictif émulé par un logiciel.

30 Une « perceuse » est une employée du centre de calcul. Sans paraître sexiste, il faut avouer que cette fonction est toujours occupée par une femme à l'époque. Elle perce (fait des trous dans) les cartes en se servant d'une énorme machine bruyante, qui s'appelle une « perforatrice ».

31 Roger Mohr a été professeur à l'École des Mines de Nancy, puis à l'Ensimag de Grenoble dont il deviendra le directeur (Tombre & al., 2017).

32 Il deviendra chargé de recherche au CNRS en physique à Orléans et son fils Nicolas Navet est professeur d'informatique à l'Université du Luxembourg.



Figure 4 - Une imprimante à tambour, capot ouvert
Source : Wikimedia Commons.

perforé s'apparente à un sifflement. Puis vient le *tacatac* régulier de la lecture des cartes perforées³³. Zut, il y a une erreur ! Nous discutons pour en trouver l'origine et proposer une suggestion de correction. Quand nous la localisons, nous utilisons la perforatrice mise à la disposition du public pour perforer une ou plusieurs nouvelles cartes. C'est reparti jusqu'à la prochaine erreur ! Vers onze heures du soir, par inadvertance, je tape sur le clavier de la console (l'IBM à boule). L'ordinateur se plante et le processus que nous avons entamé vers 19h, doit être recommencé (*boot*, chargement du compilateur, lecture du programme de simulation de la machine virtuelle, lecture de nos cartes, premiers calculs). Mes condisciples, après avoir râlé, préfèrent en sourire. Cela aurait pu leur arriver, à ce qu'ils prétendent. Au lieu de quitter le centre de calcul vers minuit, dans une prévision optimiste, nous abandonnons les lieux vers 2h du matin. Notre assembleur prototype traduit presque toutes les instructions. Nous sommes contents. Quelques petites mises au point, avec un meilleur créneau horaire, permettront de rendre un rapport et un projet acceptables. En ce qui nous concerne, comme nous ne sommes pas de véritables étudiants, c'est plutôt pour la gloire !

³³ Les imprimantes à tambour viendront plus tard et, trop bruyantes, seront installées dans des pièces séparées. Elles ont le *po-pom po-pom* qui les caractérise, qui devient très rythmé quand elles impriment un *dump* mémoire pour un chercheur qui n'a pas pu identifier la source de son erreur et utilise l'ultime recours pour traquer ce que nous n'appelons pas encore un *bug*.

C'est aussi à cette époque que je suis des cours de Claude Pair sur les automates finis, les langages à contexte libre, les grammaires et l'analyse syntaxique. À ce cours est associé un polycopié qu'a rédigé Alain Quéré³⁴ et qui correspond à des notes qu'il a prises lors d'un cours précédent³⁵. C'est à cette occasion que j'apprends l'importance du point-fixe en informatique³⁶. Je suis aussi des cours sur les graphes et les algorithmes de cheminement, théorie dans laquelle Claude Pair et Jean-Claude Derniane ont joué un rôle de pionniers et où aussi le point-fixe est fondamental (Derniane & Pair, 1971 ; Derniane, 2020³⁷).

Ma thèse de troisième cycle

Pour Claude Pair, les problèmes centraux de l'informatique naissante sont l'analyse syntaxique et les automates. Il

³⁴ Alain Quéré a été maître de conférences à Nancy. Il deviendra directeur de l'Unité Inria-Lorraine, puis directeur du Centre informatique national de l'enseignement supérieur (CINES) à Montpellier.

³⁵ Pair, « Notions sur la théorie des langages », *op. cit.*, 1968.

³⁶ Le point-fixe est au cœur du livre de Livercy (1968), car il est le pivot de la récursivité. Livercy est le nom d'auteur du collectif composé de Jean-Pierre Finance, Monique Grandbastien, Pierre Lescanne, Pierre Marchand, Roger Mohr, Alain Quéré et Jean-Luc Rémy [URL : http://denif.ens-lyon.fr/data/programmation_enslyon/2007_sem2/biblio/Livercy.pdf].

³⁷ Voir aussi Pauline Bolignano et Thierry Viéville, « La découverte scientifique... qui mérite le mérite ? », *Binaire – Blog sur lemonde.fr*, juillet 2020 [URL : <https://www.lemonde.fr/blog/binaire/2020/07>].



Figure 5 - La couverture de ma thèse de troisième cycle et la citation d'Éluard

s'intéresse aussi aux arbres (syntaxiques) dont la structure ressemble par certains aspects aux chaînes de caractères, en les généralisant, et qui jouent un rôle fondamental en analyse syntaxique. On doit pouvoir donc décrire des automates d'arbres et définir des langages d'arbres reconnus par des automates d'arbres, que l'on appelle langages reconnaissables. D'autre part, un théorème dit que, pour les chaînes de caractères, les langages reconnaissables par automates sont aussi ceux qui sont engendrés par les opérations algébriques de bases sur les langages dont la fameuse « opération étoile » ; pour cette raison on les appelle « langages réguliers ». Ils sont

aussi ceux qui sont solutions d'équations d'un certain type et pour cela ils sont algébriques. Claude³⁸ veut donc que je généralise ce résultat aux arbres. Mais il se trouve que je suis tombé sur des articles qui proposent une généralisation encore plus grande, en permettant d'énoncer ce théorème sur des structures plus générales, à savoir des algèbres abstraites, dont les chaînes de caractères et les arbres ne sont que des cas particuliers et plus généralement encore en prenant comme cadre la théorie mathématique des catégories, si bien que, par ironie, je

38 À l'époque, je ne me serais pas permis de l'appeler par son prénom.

mets en exergue de ma thèse (1971) un poème d'Éluard³⁹, avec en particulier les trois extraits « *une théorie d'arbres* [...] *nous frappe et nous divise* », « *Le chapitre inexplicable est devenu incompréhensible* » et « *Prenons la main de la mémoire* »⁴⁰ qui s'appliquent bien, je pense, à mon travail. De fait, je peux généraliser l'étoile et démontrer un théorème d'équivalence des reconnaissables, des réguliers et des algébriques⁴¹. Ce résultat n'a pas d'écho. Plus tard, en 1976, je publie en 1976 (en français), dans la revue d'informatique française *RAIRO Informatique théorique*, un article qui n'est cité, d'après le moteur de recherche d'articles académiques Google scholar⁴², que trois fois. Ainsi va la recherche en informatique française dans les années 1970 ! En arrivant au MIT en 1980, je constate que la recherche française en informatique est insignifiante pour nos collègues américains, tandis qu'en France, en informatique théorique, il existe une rivalité Paris-Province.

En 1971, c'est le moment pour moi de faire mon service militaire. Étant agrégé, je peux bénéficier de mon statut

pour l'effectuer comme professeur de mathématiques à l'École de Maistrance de l'Aéronautique navale à la base de Fréjus-Saint-Raphaël. Je suis logé sur la base et pour aller de ma chambre à la plage, il me faut contourner le tennis, ça aurait pu être pire ! Dans cette ambiance fréjus-sienne qui mélange le loisir et la Marine, ma famille me manque beaucoup, mais je souffre aussi de l'isolement scientifique. Je comble cela en continuant une recherche à la suite de ma thèse, mais celle-ci s'élève dans l'abstraction. Claude Pair en particulier m'envoie des documents à lire, en l'occurrence une thèse à évaluer (figure 6).

Heureusement, il est prévu une école d'été d'informatique (la deuxième du genre) à Neuchâtel en Suisse. J'ai raté la première édition, qui a eu lieu à Alès et je le regrette⁴³. Au cours de cette école, au demeurant très sympathique et familiale, les jeunes chercheurs sont invités à présenter leurs travaux récents. J'expose les miens, mais je ne sais pas bien m'expliquer, je ne donne aucune intuition, je m'envole dans l'abstraction. Éluard a raison et, au premier rang de l'assistance, l'un de mes meilleurs amis pleure (littéralement) de rire, tant ce que je dis est abscons. Il me faut atterrir.

Au même moment, c'est-à-dire en 1972, je rencontre une théorie qui va jouer un rôle important dans ma recherche ultérieure, ainsi que dans mon

39 Paul Éluard, « Silence de l'Évangile » dans le recueil, *Mourir de ne pas mourir*. Le titre de cet article, « Prenons la main de la mémoire », a été extrait de ce poème.

40 Ici au sens informatique du terme.

41 Claude Pair, « Colloque en l'honneur de Pierre Lescanne », Archives de l'auteur, 29 mai 2006, LORIA, Nancy.

42 N.D.E. : Moteur de recherche spécialisé en citations académiques. Ses résultats sont limités par l'ensemble d'articles de revues indexés dans sa base de données.

43 J'assisterai à celle de Tarbes en 1974 et à celle de Rabat en 1975.

UNIVERSITÉ DE NANCY

FACULTÉ DES SCIENCES

MATHÉMATIQUES APPLIQUÉES

2-1-10

Mon Cher Lescaux,

Un Grandis s'étant intéressé aux ramifications (en un d'une lettre de 3^e cycle), on me demandait mon avis sur son travail. Puis: y a-t-il une vraie chose? J'ai cependant survolé ce texte. Il me semble que beaucoup de points ne sont qu'un habillage un peu différent de ce qui avait été fait Aréole. Cependant, ce que l'on appelle grammaires régulières, et fait. et les traductions de ramifications, semblent plus intéressantes. Cela se peut être en rapport avec ce que vous avez fait.

J'attends de vous un rapport sur ce travail me indiquant ce qui vous paraît neuf et intéressant, et si les démonstrations sont correctes. Nous pourrions aussi en parler si vous venez bientôt à Nancy.

Vos vœux que j'ai écrits par écrit de... vos vacances.

Bien à vous,


 C. PAIR

107 Bd Charlemagne

Figure 6 - Une lettre de Claude Pair, reçue pendant mon service militaire

enseignement⁴⁴. En effet, je découvre le livre de Peter Wegner intitulé *Programming languages, information structures, and machine organization* (1968) qui, entre autres choses, me fait connaître le lambda-calcul. Pour compléter mon initiation, je fais acheter par la bibliothèque de mathématiques le petit fascicule dactylographié de Hindley, Lercher & Seldin (1972), qui, bien que ce ne soit pas dans son titre, présente plus complètement le lambda-calcul. C'est aussi à ce moment que Claude Pair rapporte d'une réunion du groupe de travail de l'IFIP⁴⁵, les photocopies des transparents de Dana Scott⁴⁶ sur le modèle du lambda-calcul que ce dernier vient d'inventer (Scott, 1970). Je suis fasciné et pour faire partager mon enthousiasme, j'écris, en français, pour mes collègues, une petite note sur le sujet (Lescanne, 1973). Je ne renouerai avec le lambda-calcul qu'en 1993 (Lescanne, 1994).

Pendant mon service militaire, j'ai été promu maître assistant, puis nommé à l'Université Nancy 2, l'université de lettres et droit. Je continue à enseigner les mathématiques, cette fois-ci aux psychologues. Quant au contenu, c'est assez cool et avec mon collègue Jean-Pierre Deschaseaux, enseignant en mathématiques,

nous essayons d'intéresser les étudiants, qui sont réfractaires aux mathématiques. Nous nous investissons donc beaucoup dans la pédagogie, mais lorsqu'un collègue enseignant de psychologie se lamente du grand nombre d'étudiants eu égard aux perspectives professionnelles et nous invite à être en pointe pour la sélection, car les maths sont un bon outil pour cela, alors que la psychologie en tant que discipline d'enseignement ne peut pas jouer ce rôle, nous opposons un refus catégorique. Sans le lui dire, mais en le faisant savoir aux étudiants, nous mettons au point un calcul savant de coefficients, pour qu'un étudiant qui a assisté à tous ses partiels, sans cependant remettre jamais une copie complètement blanche, soit assuré d'avoir la moyenne. Un jour que les étudiant(e)s sont en grève, je vais prendre un café avec le piquet de grève et nous discutons de tout et de rien, notamment de l'intérêt des mathématiques dans le cursus de psychologie. Une étudiante qui, je le comprends bien, n'est pas enthousiasmée par mes cours, me le dit de façon cachée : « *Oh Monsieur vous devriez assister au cours d'histoire du cinéma, c'est formidable !* »⁴⁷. Je n'enseigne toujours pas d'informatique, ni quelque chose qui y ressemble, mais un dossier pour la création d'une MIAGE⁴⁸

44 À l'École normale supérieure de Lyon, à partir de 1997, j'enseignerai le lambda-calcul à des générations d'élèves.

45 L'IFIP (International Federation for Information Processing) regroupe au niveau international les sociétés savantes en informatique.

46 Dana Scott est grand logicien américain qui s'est intéressé aux fondements de l'informatique.

47 Le cours est enseigné par Roger Viry-Babel, un an plus âgé que moi, avec qui je ne prétends pas rivaliser sur la forme comme sur le fond. Grand cinéophile, Roger Viry-Babel (1946-2006) fut aussi cinéaste, journaliste et pionnier de l'enseignement de l'audiovisuel à l'université.

48 Maîtrise d'informatique appliquée à la gestion des entreprises.



Figure 7 - Une machine à écrire IBM à boule
Source : Etan J. Tal via Wikimedia Commons.

à Nancy est déposé. On me demande d'écrire le programme de mathématiques et je m'y colle, puis je me retrouve à l'enseigner devant la première promotion : graphes et langages entre autres, mon enseignement se rapproche de l'informatique, puisque j'aborde enfin les structures mathématiques de l'informatique. Dans la plaquette de présentation j'essaie de trouver les mots qui puissent motiver des étudiants de sciences économiques à rejoindre notre nouvelle formation⁴⁹.

L'enseignement des mathématiques doit se fixer trois objectifs :

- Donner des outils qui serviront soit à d'autres enseignements, soit dans la formation permanente, soit dans la vie professionnelle.
- Introduire la démarche algorithmique et la modélisation de situations concrètes.
- Développer l'esprit de rigueur dans la formulation et la résolution des problèmes.

⁴⁹ Les enseignants de la Miage. MIAGE, « Maîtrise de méthodes informatiques appliquées à la gestion »,

Plaquette de présentation, 1973-1974, 1973. Archives de l'auteur.

On voit que plus que le contenu, c'est la démarche qui compte. C'est aussi une initiation à l'algorithmique.

En parallèle, je donne des cours au département d'informatique de l'IUT de Nancy, toujours des cours de mathématiques, toujours dans le même esprit, mais qui incluent l'algèbre de Boole et les fameux diagrammes de Karnaugh, que j'apprends pour l'occasion.

Un premier constat

À l'issue de cette période, j'ai conscience de ne pas savoir assez d'informatique. Certes, le sujet me plaît, mais j'ai surtout enseigné des mathématiques et n'ai fait que des recherches assez théoriques et formelles. D'ailleurs cette conscience de notre inculture est un constat commun parmi les collègues et a conduit à créer l'école d'été de l'« Association française pour la cybernétique économique et technique » (AFCET). La décennie qui vient me fera aborder une certaine forme d'informatique, où je mettrai les mains dans le cambouis ou plus exactement sur le clavier. Mais c'est une « informatique à la française », qui, avec ses pauvres moyens, est décalée et isolée par le plan Calcul, mais qui est pleine de promesses et qui révélera que le bon niveau mathématique des chercheurs français est un atout.

Bibliographie (citée) de l'auteur

Lescanne P. (1971). « Étude de quelques théories des langages et généralisation du théorème de Kleene ». Thèse de Spécialité, Université Henri Poincaré – Nancy 1, Juin 1971.

Lescanne P. (1973). « Introduction au lambda-calcul » [en ligne]. Site personnel de Pierre Lescanne [URL : <http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/lambda.html>].

Lescanne P. (1976). « Équivalence entre la famille des ensembles réguliers et la famille des ensembles algébriques ». *RAIRO Informatique Théorique et applications*, 10(8), pp. 57-81.

Lescanne P. (1994). « From lambda-sigma to lambda-epsilon a journey through calculi of explicit substitutions ». In H.-J. Boehm, B. Lang & D. M. Yellin (eds.). *Conference Record of POPL'94 : 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Portland, Oregon, USA, January 17-21). ACM Press, pp. 60-69.

Lescanne P. (1996). « La science informatique ». In G. Grignon (dir.). *Encyclopédie Illustrée de la Lorraine. Histoire des Sciences et des Techniques*. Metz : Éditions Serpenoises, pp. 105-116.

Créhange M., Lescanne P. & Quéré A. (2021). « L'informatique de Claude Pair ». *Bulletin de la société informatique de France*, 1024 (à paraître).

Bibliographie générale

Cohen P. J (1960). « On a conjecture of Littlewood and idempotent measures ». *American Journal of Mathematics*, 82(2), Apr., pp. 191-212.

Créhange M. & Haton M.-C. (2014).

« L'informatique universitaire à Nancy : un demi-siècle de développement ». *Technique et Science Informatiques*, 33(1-2), pp. 127-141.

Derniame J.-C. (2020). « À propos du cheminement dans les graphes » [en ligne]. *Bulletin de la société informatique de France*, 1024/16, novembre [URL : https://www.societe-informatique-de-france.fr/wp-content/uploads/2020/11/1024-numero-16_Article19.pdf].

Derniame J.-C. & Pair C. (1971). *Problèmes de cheminement dans les graphes* (Monographies d'informatique/AF CET). Paris : Dunod.

Handley J. R., Lercher B. & Seldin J. P. (1972). *Introduction to Combinatory Logic* (vol. 7 of London mathematical Society, Lecture note series). Cambridge (UK) : Cambridge University Press.

Amemiya I. & Ito T. (1964). « A simple proof of the theorem of P.J. Cohen ». *Bull. Amer. Math. Soc.*, 70(06), pp. 774-776.

Knuth D. E. (1972). « Ancient Babylonian Algorithms ». *Commun. ACM*, 15(7), pp. 671-677.

Le Lionnais F. (dir.) (1948). *Les Grands Courants de la pensée mathématique*. Marseille : Cahiers du Sud.

Legras B. (2008). *Jean Legras - Mathématicien lorrain, précurseur de l'informatique à Nancy, fondateur de l'Institut universitaire de calcul automatique*. Laxou : Groupe Dialog'Guyo (auto-édition).

Legras J. (1963). *Précis d'analyse numérique*. Paris : Dunod.

Legras J. (1963). *Initiation à l'analyse numérique*. Paris : Dunod.

Livercy C. (1978). *Théorie des programmes*. Paris : Dunod.

Mounier-Kuhn P.-É. (2010). *L'Informatique en France, de la seconde guerre mondiale au plan Calcul. L'émergence d'une science*. Paris : PUPS (2021).

Pair C. (1990). « CRIN. The history of a laboratory ». *Annals of the History of Computing*, 12(3), July / September, pp. 159-166.

Rice H. G. (1965). « Recursion and iteration ». *Commun. ACM*, 8(2), pp. 114-115.

Scott D. (1970). « Lattice-theoretic models for the lambda calculus ». *IFIP WG 2.2, Bulletin n° 5*.

Tombre K., Quan L., Horaud R., Gros P., Schmid C. & Sturm P. (2017). « In Memoriam Roger Mohr ». *Bulletin de la société informatique de France*, 1024/11, September, pp. 91-98 [URL : <https://hal.inria.fr/hal-01598085>].

Wegner P. (1968). *Programming languages, information structures, and machine organization*. New York : McGraw-Hill.

Annexe A - Glossaire

Adresse : La mémoire d'un ordinateur est divisée en emplacements qui ont chacun une adresse.

Adressage indirect : Si une instruction identifie un emplacement de la mémoire, où chercher une information, non pas explicitement, mais par l'adresse d'un autre emplacement de la mémoire qui lui-même contient l'adresse de l'emplacement recherché, on parle d'adressage indirect.

Algèbre de Boole : Une structure algébrique qui sert de fondement mathématique à l'architecture des ordinateurs.

Analyse syntaxique : Opération qui consiste à examiner une chaîne de caractères pour en extraire la structure en composants, en vue d'une traduction.

Algorithme : Ordonnancement des calculs en vue de résoudre un problème. Un algorithme est mis en œuvre par un programme.

Algorithmique : Science qui étudie les algorithmes.

Arbre syntaxique (ou arbre) : Structure vers laquelle l'analyse syntaxique traduit une chaîne de caractères.

Assembleur : Programme, relativement simple, qui traduit un programme en langage d'assembleur en un programme en langage machine.

Automate : Dispositif (programme) qui traite de l'information.

Booter : Démarrer un ordinateur en lui indiquant la première instruction qu'il doit effectuer. Sur un ordinateur d'aujourd'hui, cela est fait par une instruction à laquelle l'utilisateur lambda n'a pas accès.

Branchement : Rupture dans l'enchaînement séquentiel des instructions.

Calcul d'instruction : Un programme qui s'exécute est amené à répéter plusieurs fois la même instruction ou presque. Le calcul d'instruction est une technique absconse et obsolète qui consiste à mettre en œuvre ce « ou presque » et à modifier les instructions en programme au vol. Cela se fait en considérant les instructions du langage machine comme des valeurs numériques sur lesquelles on peut calculer.

Compilation : Action de traduire un programme dans un langage évolué (*cf.* tableau de l'annexe C) vers un programme du langage d'assembleur d'une machine.

Compilateur : Programme qui effectue la compilation.

Diagramme ou table de Karnaugh : Méthode graphique pour trouver ou simplifier une fonction logique à partir de sa table de vérité.

Éditeur : Logiciel permettant de créer et modifier un fichier. Dans un éditeur ligne à ligne, les commandes s'appliquent à chaque ligne et le résultat n'est pas immédiatement visible. Dans un éditeur plein écran, un curseur permet de voir où se fait la modification qui est immédiatement visible.

Étoile : En théorie des langages, opération sur un langage qui produit un autre langage. Par exemple l'étoile du langage $\{ab\}$, à un seul mot, s'écrit $\{ab\}^*$ et est $\{\epsilon, ab, abab, ababab, \dots\}$.

Grammaire : Ensemble de règles décrivant mathématiquement la structure d'un langage formel. Les grammaires les plus fréquemment étudiées par les informaticiens, sont les grammaires qui définissent les langages à contexte libre.

Graphe : Structure faite de points reliés par des traits sur laquelle on peut exécuter de jolis algorithmes.

Instruction : La composante de base d'un programme.

Job : Tâche qui doit être exécutée par un ordinateur. En français, tâche.

Lambda calcul : Formalisme inventé par le logicien Alonzo Church pour abstraire la notion de calcul. Le lambda calcul a des liens avec la machine de Turing, mais il lui est légèrement antérieur et est plus mathématique.

Langage : En théorie des langages, un langage est un ensemble de mots ou un ensemble d'arbres. Parmi les langages, on distingue des langages reconnaissables, des langages réguliers, des langages algébriques.

Langage d'assembleur : Langage de très bas niveau qui représente le langage machine sous une forme lisible par un humain.

Langage machine : Langage dont les instructions sont des mots machines (donc une suite de bits) qui est fait pour être interprété directement par la machine ou le processeur. Il est presque totalement incompréhensible (directement) pour un humain.

LOAD : Instruction de chargement d'une valeur dans un registre depuis un emplacement de la mémoire.

Mot : En théorie des langages, un mot est un ensemble de lettres.

Mot clé : Dans un langage de programmation, certains mots ont un usage réservé et ne peuvent pas être utilisés comme variable, par exemple *if, then, else, while, etc.*

Mot vide : En théorie des langages le mot vide est le mot qui n'a aucune lettre. Il s'écrit

aujourd'hui ϵ . Claude Pair l'écrivait 'L, probablement parce que c'était plus facile à écrire avec une machine à écrire.

Point-fixe : Solution d'une équation de la forme $x=f(x)$.

Précession (gyroscopique) : Particularité d'un gyroscope de se déplacer dans une direction perpendiculaire à son axe de rotation.

Programmation fonctionnelle : Façon d'aborder les programmes comme des fonctions à évaluer. Le programmeur ne donne pas l'organisation des calculs mais laisse le soin au compilateur de le faire.

Récursivité : Propriété d'un programme d'être récursif.

Récursif : Un sous-programme (une procédure) est récursif (récursive), s'il (elle) contient dans son code un appel à lui-même (elle-même).

Réécriture : (Pour faire simple) la réécriture est la théorie de la simplification des formules.

Registre : Pour accélérer le calcul, un ordinateur possède des emplacements de mémoire à accès plus rapide que le reste de la mémoire. Ces emplacements spécifiques sont appelés des registres.

Régulier : En théorie des langages, un langage est régulier s'il peut être engendré par des opérations simples, à savoir l'union, la concaténation et l'étoile.

Switch : Minuscule levier qui a deux positions, haut ou bas.

Système d'exploitation : Logiciel de base d'un ordinateur, qui donne accès à la machine, gère ses entrées et sorties, permet à plusieurs personnes de l'utiliser et fait fonctionner plusieurs processus en même temps.

Tampon : Mémoire provisoire sur laquelle sont enregistrées les données sur lesquelles l'utilisateur interagit. En anglais, *buffer*.

Temps partagé : Quand plusieurs utilisateurs ou plusieurs processus peuvent s'exécuter en même temps on parle de temps partagé, en anglais *time sharing*.

Traitement par lot : Quand les programmes exécutés par un ordinateur sont exécutés les uns après les autres, on parle de traitement par lots, en anglais *batch*. Le contraire est le temps partagé.

Théorie des catégories : Théorie mathématique ayant un haut niveau d'abstraction.

Triangle de Pascal : Méthode inventée par Blaise Pascal pour calculer des nombres, à savoir le nombre de combinaisons de n éléments p à p .

Annexe B - Plan du polycopié « Initiation à l'Algol »

- Chapitre I : Généralités
- Chapitre II : Notions permettant d'écrire des programmes simples
- Chapitre III : Nombres, Variables, Expressions arithmétiques simples, Fonctions, Affectations
- Chapitre IV : Les instructions d'entrées et de sorties
- Chapitre V : ALLERA ; Instructions conditionnelles et composées
- Chapitre VI : Les ordres de bouclage : Instruction POUR
- Chapitre VII : Expression booléennes
- Chapitre VIII : Blocs et déclarations
- Chapitre IX : Les procédures
- Chapitre X : Compléments : Expression conditionnelle, Aiguillage, Expression de désignation, commentaire, La partie valeur, Instruction vide
- Annexe
- Exercices d'Algol
Parmi les exercices, j'ai noté que ce que nous appelons aujourd'hui un tri est appelé un classement des éléments d'un tableau par valeurs croissantes (Exercice 5.4).

Annexe C - Quelques langages de programmation

FORTRAN	1957	Premier langage de programmation
ALGOL 60	1960	Premier langage avec récursivité et structure de blocs
COBOL	1959	Langage pour la gestion
SYMBOL	1968	Langage d'assembleur du CII 10070
PASCAL	1970	Langage issu d'ALGOL 60, avec des structures de données et une recherche de la simplicité

Une pratique active de l'informatique à l'école

Gilles Blain

Maître de conférence honoraire à Sorbonne Université

Gérard Nowak

Maître de conférence honoraire à Sorbonne Université

Didier Vaudène

Maître de conférence honoraire à Sorbonne Université

Résumé

De 1980 à 1986, à l'initiative d'universitaires de l'Institut de programmation (Université Paris VI), a été élaborée et mise en œuvre une expérience avec l'environnement Logo dans un groupe scolaire parisien, concernant les enfants dès la grande section de maternelle. L'objectif principal n'était pas d'utiliser l'informatique comme un outil, mais de la pratiquer, tout en favorisant la collaboration et l'interaction dynamiques entre les enfants. Les activités Logo étaient intégrées (comme activités d'éveil) et assurées par des instituteurs volontaires auxquels nous avons donné une formation. Nous rappelons le contexte et les étapes essentielles de l'expérience, tout en proposant diverses remarques suggérées par notre regard d'informaticiens.

Mots-clés : : micromonde Logo ; pédagogie par projets ; dynamique de groupe ; analyse réflexive des erreurs ; programmation informatique ; effectivité des machines et des interprètes.

Abstract

From 1980 to 1986, due to the initiative taken by teachers-researchers of the Institut de Programmation (Université Paris VI), a Logo experience was elaborated and implemented in a primary school in Paris (children from 5 to 10 years old). The main goal was not to introduce computers as tools, but to learn and practice programming, while encouraging dynamic collaboration and interaction between children. Logo activities were integrated (as creative ones) and took on by voluntary teachers to whom we had given a first-degree course. We recall the historical context and the main steps of the experience, and we propose some remarks suggested by our computer scientists' point of view.

Keywords: Logo microworld; project-based learning; group dynamics; reflexive error analysis; computer programming; effective machines and interpreters.

« *Les enfants doivent être très indulgents envers les grandes personnes.* »

Antoine de Saint-Exupéry
Le Petit Prince, 1943.

Introduction

Comme chaque année, à la fin du second semestre, l'association de parents d'élèves affiliée à la Fédération des conseils de parents d'élèves des écoles publiques (FCPE) du Groupe scolaire Corbon-Alleray, dans le 15^e arrondissement de Paris, organise une kermesse où les élèves peuvent présenter leurs travaux et participer à des activités ludiques dans une journée conviviale qui réunit les enfants, les parents et les enseignants. En 1980, un enseignant de l'Institut de programmation (IP), Gilles Blain, dont les enfants sont scolarisés dans cette école, est président de cette association. Et quand il apprend qu'un de ses collègues, Gérard Nowak, a déjà co-organisé des présentations de Logo¹, aussi bien à des collégiens, lors d'un camp d'été², qu'au grand public, lors d'une exposition sur la micro-informatique au

Palais de la découverte, il ne reste plus qu'à vaincre les obstacles administratifs et matériels pour transporter et installer, dans l'une des classes de l'école, le temps d'une journée, un mini-ordinateur de l'IP équipé d'un système Logo. « *Mais pourquoi s'en tenir à une seule journée ?* » insiste Nowak devant l'intérêt que rencontre cette initiative lors de la kermesse. Comme une sorte de défi, ce qui ne devait être initialement qu'une activité ludique d'une après-midi d'été est devenu dès la rentrée 1980 une expérience multiforme, pionnière à Paris, qui s'est poursuivie jusqu'en 1986, quand le plan Informatique pour tous aura été mis en place au niveau national et prendra le relais. Cette expérience a concerné tous les niveaux, depuis la grande section de maternelle jusqu'au cours moyen deuxième année³. Elle a mobilisé les énergies (et même parfois les inerties) des acteurs et des instances mis en jeu par l'organisation d'une telle expérience, tant du côté universitaire, pour la formation des maîtres et le prêt de certains matériels, par exemple, que du côté académique, rectoral, municipal et scolaire pour les autorisations administratives, la sécurité et quelques apports budgétaires en particulier.

Qu'est-ce qui a précédé cette étincelle et l'a en quelque manière rendue possible ? Quels objectifs et quelles

¹ Environnement d'apprentissage, conçu à la fin des années 1960, utilisant des micro-ordinateurs et favorisant les interactions et les collaborations entre les enfants.

² IRIA (dir.) (1979). Camp d'été Jeunes et Ordinateurs. Rapport d'expérimentations pour le Ministère de l'industrie (Universités de Paris VI, Paris VIII, Le Mans et Dijon). L'IRIA était l'Institut de recherche en informatique et automatique.

³ Dans la suite du texte, les acronymes suivants indiqueront : GSM : Grande section de maternelle ; CP : Cours préparatoire ; CE : Cours élémentaire ; CM : Cours moyen.

idées ont guidé, avec quelque obstination parfois, la mise en place d'une telle expérience dès la grande section de maternelle et à tous les niveaux du primaire ? Comment une expérience pédagogique acquise par quelques enseignants au niveau d'enseignements universitaires a-t-elle été adaptée au niveau du primaire ? Comment est-elle intervenue dans la concertation avec les maîtres et les maîtresses (à vrai dire, seulement les maîtresses) qui ont souhaité participer à cette expérience et la prendre en charge auprès des enfants ? Comment une institution comme une école primaire qui, à l'époque, n'était pas prévue – c'est le moins qu'on puisse dire – pour cette sorte d'activité, a-t-elle accueilli cette expérience ? Quels ont été les rapports avec les autorités de tutelle ? Quels ont été les prolongements dans le domaine de la recherche ? Quel regard rétrospectif pouvons-nous porter aujourd'hui sur cette expérience, et surtout un regard d'informaticiens qui ne peut être ni celui de l'historien, ni celui du didacticien ? Autant de questions, de thèmes et de perspectives dont le tissage est ici résumé synthétiquement comme l'expérience d'une pratique active de l'informatique à l'école.

Un contexte favorable

C'était il y a une quarantaine d'années. On ne saurait revenir sur cet événement sans rappeler certains aspects du contexte, proche ou lointain, dans lequel

il a eu lieu. Si plusieurs acteurs peuvent ici intervenir directement, autant à travers leurs souvenirs et leurs témoignages que par les documents qu'ils ont conservés dans leurs archives, d'autres se trouvent éloignés ou ne sont plus joignables, ou connus seulement à travers des archives ou des documents publiés auxquels nous avons pu avoir accès. Et comme tout événement, à mesure qu'on l'approche plus en détail, révèle un tissage dense dont les fils et les ramifications sont illimités, c'est avec quelque arbitraire que certains traits seront soulignés, d'autres estompés ou laissés dans l'ombre, tandis que d'autres encore nous sont demeurés insoupçonnés. Il ne peut donc s'agir ici de brosser un panorama exhaustif, mais de proposer quelques repères se trouvant sur le plan de coupe où cette expérience croise ce contexte.

Les années 1970 voient l'essor de plusieurs gammes de mini-ordinateurs à un coût permettant une assez large diffusion pour l'époque, aussi bien dans le domaine industriel et commercial que dans celui des universités et des laboratoires de recherche (PDP-11 de DEC, Mitra 15 de CII-HB, T1600 et Solar 16 de Télémécanique, etc.). Comparées aux machines actuelles, les configurations sont encore modestes, surtout en mémoire centrale, et les processeurs ne sont pas très rapides. Mais elles permettent déjà une grande variété d'expérimentations et de développements dans les domaines les plus divers, dont ceux de l'intelligence artificielle et de l'éducation. Si le premier micro-ordinateur date de 1973 (le Micral

N de R2E, construit par François Ger-nelle), le PC⁴ d'IBM ne sera présenté à la presse qu'en août 1981.

En France, l'informatique⁵ se développe dans le cadre du plan Calcul (à partir de 1966), puis dans le cadre du VI^e plan (à partir de 1971). Les universités et les laboratoires de recherche s'équipent progressivement. Au début des années 1970, les pouvoirs publics prennent conscience d'un phénomène de société qui conduit à considérer que l'informatique doit être introduite dès l'enseignement secondaire, ce qui se prolongera, dans les années 1980, par une introduction de l'informatique dans l'enseignement primaire, non sans diverses errances, changements de caps et interruptions qui dessinent un zigzag chaotique et politique. C'est ainsi que se met en place, de 1970 à 1976, l'expérience dite des « 58 lycées », ne visant pas l'enseignement comme tel d'une nouvelle discipline, mais plutôt une imprégnation de ces technologies nouvelles dans une approche où chaque discipline utilise ces technologies pour valoriser son contenu

4 Le *personal computer* (PC) est le premier micro-ordinateur d'IBM dont l'architecture, délibérément non brevetée, s'est *ipso facto* imposée comme une norme de base dont dérivent toujours les PC actuels.

5 À l'époque, l'appellation « le numérique » n'existait pas dans le sens générique et passe-partout où on l'emploie depuis quelques années. C'est une difficile question de déterminer dans quelle mesure on peut (ou non) soutenir que « le traitement (éventuellement automatique) de l'information » et « le numérique » peuvent être considérés comme équivalents, voire même synonymes. La référence à l'« informatique » a été maintenue ici dans le souci d'éviter les anachronismes.

propre, et qu'on peut, dans son principe, comprendre comme de l'enseignement assisté par ordinateur (EAO). Les promoteurs en sont principalement les professeurs Jacques Hebenstreit (Supélec) et Jacques Arsac (Université Paris VI). Un langage est créé, le LSE (langage symbolique pour l'enseignement), destiné au développement, par des enseignants, de contenus favorisant l'apprentissage dans différents domaines enseignés dans les lycées. Ce n'est que dans un deuxième temps que l'apprentissage de l'informatique et de la programmation est envisagé avec une adaptation de LSE. C'est ensuite, en 1978-1980, le plan « 10000 micros » pour les collèges et les lycées, puis plusieurs opérations de 1981 à 1985, dont le plan « 100000 micros » et enfin le plan Informatique pour tous (IPT), annoncé en janvier 1985, mis en œuvre dès la rentrée de septembre 1985, mais interrompu et abandonné en mars 1989.

Ces années 1970 voient aussi l'essor, au sein du Laboratoire d'Intelligence Artificielle (AI Lab) du Massachusetts Institute of Technology (MIT), de travaux menés conjointement par Marvin Minsky, collègue de John McCarthy, auteur du langage Lisp (*list processing*), Seymour Papert (qui crée le laboratoire Logo en 1970) et la société Bolt, Beranek & Newman (BBN) pour le développement du langage Logo, et particulièrement son implémentation sur mini-ordinateurs. Les premières tortues de sol⁶ Logo voient le

6 La tortue de sol est un engin traceur sur roulettes, muni d'un « crayon », dont les déplacements et girations

jour en 1971. Papert venait de travailler plusieurs années en étroite collaboration avec Jean Piaget à Genève sur les processus cognitifs en jeu dans la construction et l'acquisition des connaissances, particulièrement chez les jeunes enfants. De là sont nées des perspectives fécondes concernant l'apprentissage autonome, les rôles d'apprenant et de personne-ressource, la pédagogie par projet (par projets graphiques, entre autres), la banalisation de la notion d'erreur, etc. Les résultats pédagogiques attendus sont alors décrits et diffusés à travers de nombreuses publications dans les séries *AI Memo* et *Logo Memo*, en particulier le mémo « Uses of technology to enhance education » (Papert & al., 1973). Ces textes suscitent rapidement un grand intérêt au Québec⁷, ainsi qu'en France à l'Institut national de recherche pédagogique (INRP) qui suit les recherches sur Logo depuis 1968, peu de temps après la première version d'un interpréteur Logo. En 1978, l'INRP rassemble et coordonne les travaux naissants d'une dizaine de petites équipes très motivées en créant une structure de recherche coopérative sur programme (RCP) dont l'intitulé « Pratique active de l'informatique par l'enfant » est clairement entendu par chaque équipe de la RCP comme un

thème de recherche⁸. Les membres des équipes, enthousiasmés par Logo qui est en train de se diffuser en France, avaient tous déjà entrepris quelques activités liées à Logo et à Lisp (Université de Vincennes) en milieu scolaire (école, collège, lycée, formation pour adultes, coopération avec des Écoles normales) ou parascolaire (camp d'été d'Arc-et-Senans (IRIA, 1979 ; Vivet, 1979), présentations au Palais de la découverte, clubs informatiques). Les idées de Papert, développées dans son livre *Jaillissement de l'esprit* (1981), et le rôle fédérateur de l'INRP ont été déterminants, au moment opportun, pour valoriser tous les efforts jusqu'alors dispersés fournis par des collègues qui, pour beaucoup, ne se connaissaient pas. Les premiers colloques autour de Logo (CNRS⁹/LISH¹⁰, 1984) ont pu bénéficier de cette synergie, qui a en outre largement favorisé la diffusion et l'étude des recherches sur Logo. Au niveau national, une prise de conscience du lien entre informatique et éducation s'est traduite en particulier par le rapport demandé à Jean-Claude Simon en 1980 par le Président de la République, *L'éducation et l'informatisation de la société* (Simon, 1981). Grâce à plusieurs contributeurs, tels Harald Wertz (Université de

peuvent être programmés en Logo, et qui permet de réaliser des dessins sur des feuilles de papier posées à même le sol (voir fig. 5).

⁷ Dès 1971, le Québécois Guy Montpetit, qui avait étudié les théories de Piaget à Genève avec Papert, mène des expériences pionnières de Logo dans un collège de la banlieue de Montréal et collabore activement au groupe Logo du MIT (Chadwick, 1984, p. 28).

⁸ INRP (dir.) (1981). « Recherche coordonnée sur programme (RCP) – Logo ». ADI/INRP. Projet n° 455. Premier rapport intermédiaire, mai 1981. INRP (dir.) (1983). « Recherche coopérative sur programme (RCP) – Pratique active de l'informatique par l'enfant ». CNRS/INRP. Projet n° 697. Rapport d'activités de 1980 à 1983.

⁹ Centre national de la recherche scientifique.

¹⁰ Laboratoire d'informatique pour les sciences de l'homme.

Vincennes) et Martial Vivet (Université du Maine), la communauté Logo, qui a été sollicitée, a pu développer dans l'annexe II du rapport – « Apprentissage autonome » –, la proposition d'un contexte pédagogique novateur.

Est-ce un paradoxe que le faisceau de recherches qui conduit à une pratique active de l'informatique via Logo en primaire, et même en grande section de maternelle, noue initialement la psychologie et l'épistémologie de Piaget, l'intelligence artificielle, les développements innovants et un langage de haut niveau comme Lisp ? Réciproquement, si des chercheurs qui sont liés à une ou plusieurs branches d'un tel faisceau se trouvent regroupés dans un même lieu (en l'occurrence à l'Institut de programmation), il y a des chances non négligeables pour que l'étincelle s'y propage ou s'y traduise. C'est ce qui se produit à l'IP où Blain, du fait de son intérêt pour les travaux de Piaget, croise les expérimentations de Papert, tandis que Nowak suit de près ce qui se fait en Lisp¹¹ et croise ainsi le langage Logo. En 1978, un système Logo québécois (PDP-11 doté de deux écrans dont un réservé aux évolutions de la tortue) est acquis par l'IP pour des recherches liées au langage Logo. Et quand l'étincelle se produit, la volonté d'Arsac de diffuser l'informatique dans le milieu scolaire, l'intérêt de Claude Girault pour Logo, la tradition

pionnière de l'Institut, et aussi beaucoup d'enthousiasme et de persévérance parviendront à surmonter les diverses difficultés qu'on peut deviner pour monter le stand de la kermesse et entreprendre l'expérience qui suivit. Cette expérience est née d'un défi relevé par deux personnes, sans la volonté originelle d'institutions. C'était possible à l'époque, ce qui autorisait une certaine liberté dans les projets, à la condition d'accepter en contrepartie de tout organiser et de consacrer suffisamment d'énergie pour mobiliser les acteurs et les instances concernées.

Une première année... et d'autres après

À la fin des années 1970, s'il y a bien une sensibilisation aux technologies nouvelles au niveau des collèges et des lycées, il y a encore très peu d'initiatives au niveau primaire, et encore moins en maternelle. Il est vrai que la perspective d'une utilisation de ces technologies nouvelles dans le cadre des savoirs traditionnels conduit plutôt à un face-à-face solitaire de l'enfant et de la machine. Aussi n'est-il pas surprenant que nous ayons pu assister à certaines expériences menées par des écoles normales ne concernant que très peu d'élèves à la fois, souvent un ou deux, entourés par un ou deux instituteurs, un ou deux professeurs d'école normale, un ou deux inspecteurs en formation, un psychopédagogue, etc., et on comprend alors aisément que ces élèves mèneront

¹¹ En particulier les travaux de Patrick Greussay et de Jérôme Chailloux à Paris VIII sur les implémentations d'interpréteurs Lisp.

leur projet sereinement dans une dynamique propre à leur âge, et que ces observateurs attentifs ne sauraient en rien perturber ni interférer avec les « phénomènes » à observer et à mesurer !

Au contraire, cette expérience d'une pratique active est d'emblée conçue comme complètement intégrée dans l'activité scolaire, en ce sens que : (1) les machines sont installées à demeure dans l'école, à chaque fois pour une classe complète et dans la perspective d'une dynamique collective ; (2) ce sont les instituteurs qui sont formés, qui maîtrisent donc les éléments techniques et qui demeurent ainsi autonomes dans leur classe ; et enfin (3) les activités Logo sont accueillies, au sein des plannings et des programmes, comme activités d'éveil, lesquelles comprennent aussi les activités d'éducation physique et sportive (EPS), de dessin, de musique et de piscine¹². En outre, par opposition à une utilisation de moyens informatiques comme d'une sorte d'outil au service de contenus d'enseignement divers (c'est-à-dire de l'enseignement assisté par ordinateur), il s'agit de promouvoir une pratique de l'informatique mettant en jeu la collaboration et l'interaction entre les élèves, soit en demi-classes, soit en petits groupes de cinq ou six, tout en favorisant autant que possible l'articulation et la coordination de plusieurs domaines, y compris, le cas échéant, l'éducation physique et le dessin, et tout en veillant à une introduction gra-

duelle des schémas de raisonnement, des concepts et des difficultés liés à l'informatique. Les élèves produiront en général en fin d'année scolaire, un texte décrivant leurs activités Logo, leurs projets, leurs réussites ou leurs échecs, leurs critiques, leurs désirs pour le futur. Belle relation en fait entre les activités d'écriture de textes en français, postérieures à celles d'écritures de séquences en Logo. Des résultats de projets graphiques pouvaient ainsi être montrés aux parents lors de la kermesse de fin d'année.

La mise en œuvre du défi initial doit affronter – et surtout résoudre – diverses difficultés d'ordre administratif, d'encadrement et de formation, de pédagogie et de mise à disposition du matériel. Rien de tout cela n'aurait pu être concrétisé sans l'aide et le concours de deux associations de parents d'élèves, l'une affiliée à la FCPE et l'autre à l'Association des parents d'élèves de l'école publique (l'APEEP), du directeur de l'école, Monsieur Radais, et d'un noyau dur d'enseignantes volontaires pour une formation initiale leur permettant de prendre en charge les activités dans leurs classes. Tous très motivés. Toutes les contraintes qui sont apparues ont pu être levées entre la rentrée de 1980 et la fin mars 1981. Par une décision en date du 25 mars 1981, le rectorat autorisait des activités dans quatre classes, en cours préparatoire (CP), cours élémentaire 1 et 2 (CE1, CE2) et en cours moyen 1 (CM1), du 11 mai au 3 juillet, pour environ huit séances par classe. Elle était assortie de quelques conditions : présence d'un psychopéda-

¹² Dans ce contexte, les activités Logo ont occupé 2h30 sur les 6h d'activités d'éveil par semaine de 27h.

gogue d'une École normale d'instituteurs (ENI), production de rapports de suivi, contacts avec des inspecteurs, aux niveaux départemental et administratif.

Un aspect crucial de cette expérience est celui de la formation à l'informatique pour que les maîtresses qui se sont portées volontaires puissent devenir autonomes et prendre en charge les activités Logo dans leurs classes, les enseignants universitaires devenant alors de simples « personnes-ressources »¹³. Une formation initiale de quatre enseignantes de CP, CE et CM, a eu lieu chaque mercredi matin à l'université Pierre et Marie Curie (UPMC), campus Jussieu, de la rentrée 1980 à avril 1981. Il convient de souligner que toutes les personnes concernées par cette formation étaient volontaires et bénévoles, aussi bien les maîtresses qui ont prélevé ce temps de formation sur leur propre temps libre, que les enseignants de l'UPMC qui ont assuré la formation.

Peut-être convient-il aussi de resituer une telle formation dans le contexte du tout début des années 1980 : aucune de ces maîtresses n'avait approché d'ordinateur, et les « nouvelles technologies » n'avaient pas encore vu le jour dans le grand public. Corrélativement, une telle formation ne pouvait pas bénéficier d'une « ambiance numérique » qui aurait déjà propagé des idées, des pra-

tiques et des images facilitant l'accès à une pratique de l'informatique. Cette ambiance a très rapidement évolué dans le courant des années 1980. Ainsi des enseignants d'informatique de l'UPMC avaient demandé en 1982 à l'Atelier de recherche du département d'audio-visuel de Paris VII la réalisation d'un vidéogramme¹⁴ pour faciliter une prise en mains rapide des premiers micro-ordinateurs utilisés dans les travaux pratiques de DEUG (Diplôme d'enseignement universitaire général), tout en insistant pour que le principe d'utilisation d'un clavier soit détaillé, l'expérience ayant montré que nombre d'étudiants n'avaient jamais touché un clavier de machine à écrire, ni *a fortiori* un clavier d'ordinateur !

La première année, la formation des quatre enseignantes volontaires (Mesdames Benisty, Debroas, Gineste, Marquez) pendant 7 mois à raison d'une demi-journée (3h) par semaine, représente environ 70h ; il faut donc faire des choix et présenter les principes et concepts essentiels dans leurs aspects majeurs. Il s'est agi de la représentation de l'information, du codage, des notions de caractères, d'alphabets, de mots, de types de données, de lignes, de fichiers, de répertoires, de types d'opérations, de variables, de structures de contrôle (la séquence, le choix, la répétition),

¹³ Pour les deux premières années : Marie-France Le Roch, Gérard Nowak, Gérard Bossuet, Pierre Cointe et Eugen Neidl.

¹⁴ Vaudène D., Nowak G. & Cheminaud M. (1983). « Micral 9050 – Mode d'emploi ». Vidéo, 23 mn. Production : Atelier de recherche du département d'audio-visuel de l'Université Paris VII (Prise en main du Micral 9050 sous CP/M).

de traducteurs, du fonctionnement d'un processeur exécutant un programme, de sauvegarde de fichiers... Pour aborder la construction de petits projets et la représentation de leurs données, les quelques principes du monde des objets et des classes d'objets ont été présentés. Dans ce contexte, on peut facilement caractériser la classe des objets « tortue Logo » en précisant leurs propriétés (une position, une direction, l'état levé ou baissé d'un « crayon ») et les comportements qui peuvent leur être appliqués (changer de position, de direction, etc.). Aborder le micromonde¹⁵ pédagogique concernant la tortue Logo se trouvait ainsi facilité. L'accent a ensuite été mis sur l'apprentissage du langage Logo (manipulation de textes, fonctions et procédures prédéfinies, possibilités de définir d'autres fonctions et procédures, etc.).

Les aspects d'organisation ne se sont pas posés de la même manière suivant que l'expérience met en jeu une tortue de sol (GSM et CP) ou un ordinateur (CE et CM). Dans le cas de la tortue de sol, c'est toute la classe (ou un demi-groupe) qui participe collectivement à l'activité. Hormis la tortue elle-même, aucune installation supplémentaire ou assignée à une salle particulière n'est nécessaire, et les contraintes matérielles sont de ce fait allégées. Par contre, l'utilisation d'ordinateurs est plus contraignante, d'autant qu'il n'y

avait qu'un seul ordinateur les deux premières années. Au milieu des difficultés d'infrastructure liées, par exemple, à des aspects de sécurité (installer des prises de courant avec raccordement à la terre) et de fourniture du courant électrique (l'école était encore équipée en 110 volts), l'expérience Logo a cependant bénéficié du fait que l'école comportait des salles de classe inoccupées. Il a ainsi été possible d'utiliser deux salles contiguës communicantes pour Logo, l'une pour l'installation des ordinateurs (un seul les deux premières années), l'autre pour le travail de préparation et de réflexion. D'un point de vue pédagogique, outre les aspects pratiques d'une telle disposition, cette séparation permet de favoriser les moments correspondant au travail d'imagination et de préparation, s'accomplissant sans machine, les élèves étant répartis en groupes de 3 à 5, chacun choisissant son projet graphique et commençant à élaborer sa mise en œuvre sur papier. Pendant ce temps, d'autres peuvent tester des séquences de commandes sur ordinateur, et céder leur place lorsque le résultat est atteint ou lorsqu'une difficulté exige une réflexion qui ne requiert pas l'usage de l'ordinateur.

Cette expérience de Logo est aussi, pour nous, une expérience de la mise en œuvre d'une telle expérience, en particulier les deux premières années, avec les difficultés matérielles, organisationnelles, relationnelles, etc., qui peuvent surgir dans un tel contexte (Nowak, 1982). Au cours des six années, l'équi-

¹⁵ Concept inventé par Papert et Minsky, concernant les environnements de pédagogie active pour des apprentissages avec autonomie (Papert, 1980).

pement a lentement mais favorablement évolué. Après deux années scolaires vécues avec le PDP-11 prêté par l'IP, une coccinelle Logo de Roger Tanguy et un Promobile-Tortue Jeulin¹⁶, l'école a profité de février à fin octobre 1983 de 4 micro-ordinateurs Micral 8022G de R2E, prêtés par le Ministère de l'éducation nationale (MEN) et d'une seconde tortue Jeulin. De novembre 1983 à avril 1985, nous ne disposons plus que de deux Micral (les deux autres ayant été repris et réaffectés ailleurs par le MEN). En avril 1985, la ville de Paris, via la mairie du 15^e arrondissement¹⁷, équipe l'école de 4 postes Thomson TO7 dotés du langage Logo et de 2 tortues Jeulin. En 1985-1986 la dotation du rectorat et du plan IPT permet de disposer de l'équipement d'un Nano-réseau avec 6 postes TO7 et 6 postes MO5, ce qui devenait très raisonnable pour mener les activités¹⁸.

¹⁶ Les coccinelles Logo figurent parmi les premières tortues de sol Logo produites en France, en l'occurrence celles de Roger Tanguy produites dans le cadre de l'université Paris VIII. Un peu plus tard, en 1981, l'entreprise Jeulin développe une tortue de sol produite en petite série sous le nom Promobile-Tortue.

¹⁷ Il convient de souligner le rôle décisif du maire, René Galy-Dejean.

¹⁸ Le Nano-réseau, conçu et développé par l'Université des sciences et techniques de Lille (USTL) et l'entreprise Léanord, est un petit réseau d'ordinateurs composé d'une « tête de réseau » (un Sil'Z de Léanord) et de nano-postes (des nano-ordinateurs grand public TO7 ou MO5 de Thomson). La tête de réseau permet de diffuser des didacticiels aux nano-postes, de sauvegarder et de restaurer des états de projets en construction, et d'imprimer.

Projets Logo graphiques coopératifs

Dès la première année, un travail a été mené dans les classes de GSM et de CP¹⁹, avec une tortue de sol. Les élèves apprennent à connaître les actions de base de la tortue qui sont principalement organisées autour du pivotement sur place (à gauche ou à droite, d'un certain nombre de tours, de demi-tours, de quarts de tours, d'un certain angle) et du déplacement dans la direction courante (en avant ou en arrière, d'un certain nombre de pas), d'autres actions permettant de lever ou de baisser le crayon, d'allumer ou d'éteindre des diodes, et d'émettre quelques sons. À chaque action possible est associée une configuration de perforations ménagées dans des cartes en matière plastique, de la taille d'une carte de crédit, qui peuvent être insérées dans un petit lecteur approprié, lequel transmet par radio à la tortue la commande des actions à effectuer (figure 1).

Dans une première étape, en GSM et au début du CP, aucune séquence de commandes n'est enregistrée : la tortue effectue immédiatement l'action associée à une carte dès qu'elle a été insérée dans le lecteur. Il n'y a donc pas encore de schéma de contrôle (test, alternative, itération, etc.) qui soit instancié comme tel, et seules les actions élémentaires sont associées à une carte-symbole. À ce stade, s'il n'y a pas à proprement parler de pro-

¹⁹ Classes de Mesdames Lebreau, Rocabois et Villaudière.

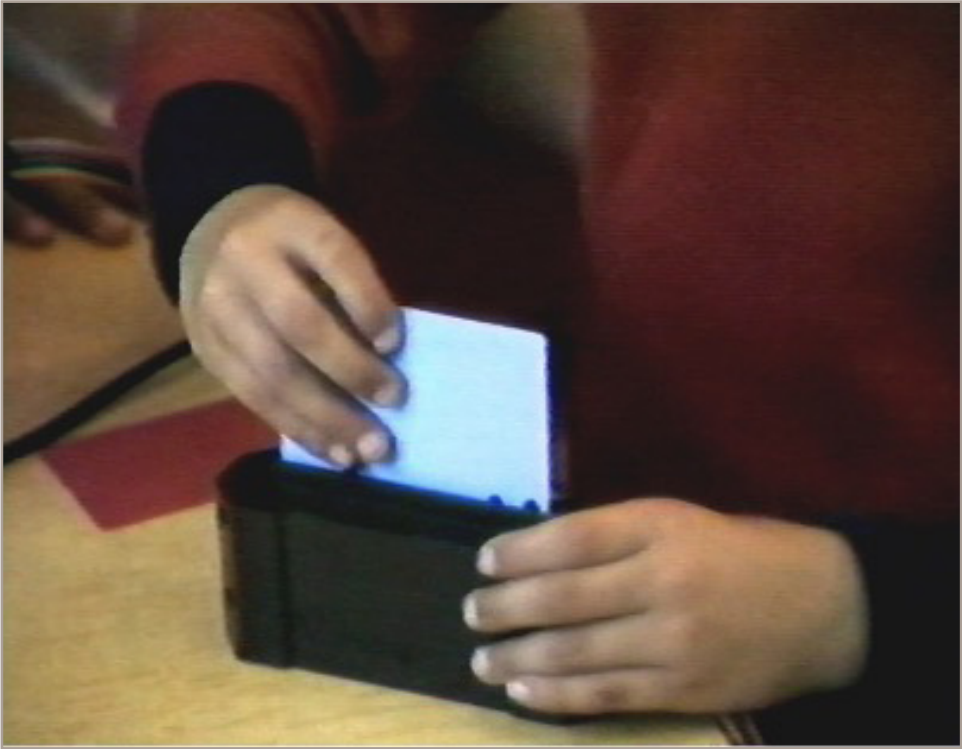


Figure 1 - Insertion d'une carte-commande dans le lecteur (GSM)

Crédit photo : Didier Vaudène.

gramme – en tant que détaché et instancié symboliquement comme tel pour être enregistrable –, il y a cependant programmation, car la programmation est effectuée, mieux, elle est vécue comme un dispositif collectif d'élaboration qui conduit les élèves à se concerter, à débattre et à décider à chaque pas quelle commande il convient maintenant de transmettre à la tortue : « *et maintenant, tu fais avance trois..., tourne à droite quart de tour...* ».

Après une étape de familiarisation avec les actions de base, chaque élève est amené à proposer un symbole gra-

phique, traduction de sa compréhension de chaque action. Des présentations de ces symboles à toute la classe, des compléments par explications d'élèves, par mimes, des discussions, doivent favoriser l'émergence d'un code graphique produit et accepté par la classe, soit un ensemble d'une dizaine de symboles graphiques qui remplaceront avantageusement les étiquettes standards retenues par le constructeur pour identifier les cartes.

Cette photographie (figure 2) montre un extrait de feuilles de papier accrochées au tableau, où chacun peut venir appor-

	allume	éteins	lève crayon	baisse crayon	avance	recule	tourne droite	tourne gauche	joue
patrick F.									
marie-milla									
claire-judith									
tamer									
estelle									
raphaël									
david									
otavia									

Figure 2 - Table des symboles graphiques
Crédit photo : Gérard Nowak.

ter ses propositions : les colonnes correspondent aux actions possibles et chacun a sa ligne.

Dans cet exemple (figure 3), la déclinaison de la commande « pivote droite » est symbolisée d'une manière qu'on pourrait dire « analogique » : plus l'angle du pivotement est grand, plus la spirale symbolisante comporte de tours. Cette attention portée à la créativité des élèves est aussi l'espoir d'un embryon d'appropriation d'une forme de langage par les élèves, de leur maîtrise sur les actions des robots (figure 4), au sein d'un travail d'abord individuel, puis surtout collectif, plus riche au plan pédagogique autant qu'à celui de la sociabilité²⁰.

La finitude n'épargne pas le micro-monde de la tortue Logo, et ce que la perfection lissée des idéalités géométriques laisse imaginer se heurte bien vite à la granularité rugueuse et empirique des tracés et de la discrétisation. Dès qu'un élève imagine qu'il pourra tracer des ronds ou des cercles, il ne pourra jamais tracer qu'une figure approchée comme un polygone régulier. En outre, quand le périple de la tortue se terminera pour un tel tracé, il faudra encore veiller à ce que le trait rejoigne le point de départ pour fermer

ont intéressé des responsables de l'ENI d'Auteuil ainsi que l'Association générale des institutrices et instituteurs des écoles et classes maternelles publiques (AGIEM). Elles ont débouché sur la mise en place de stages de formation « École maternelle, Ordinateur, Informatique » pour les enseignants de GSM, auxquels Madame Lebreau a apporté son concours. Un rapport interne sur ces formations a été diffusé en 1983 auprès des membres de l'AGIEM.

²⁰ Les expérimentations des années 1980-1982 menées par Madame Lebreau en grande section de maternelle

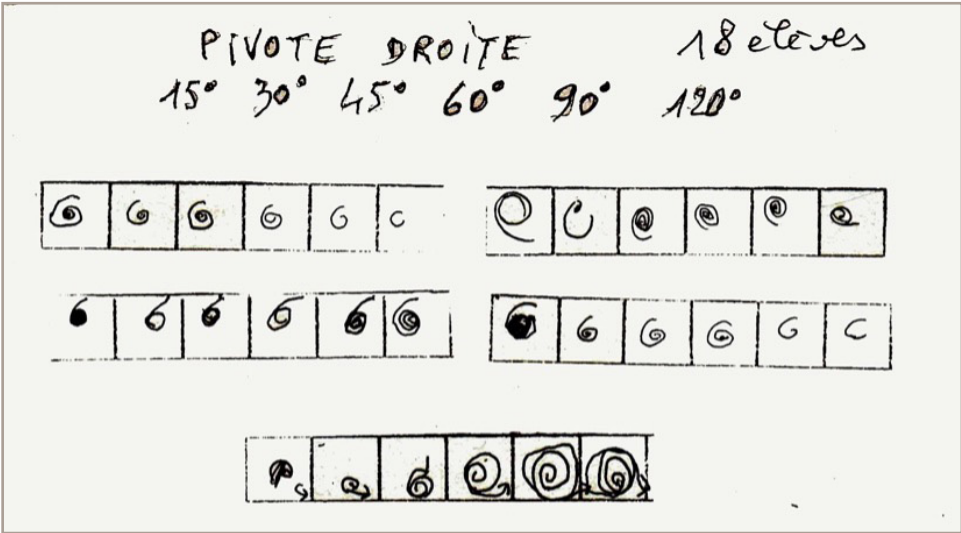


Figure 3 - Symbolisations de la commande « pivot droite »

Crédit photo : Gérard Nowak.

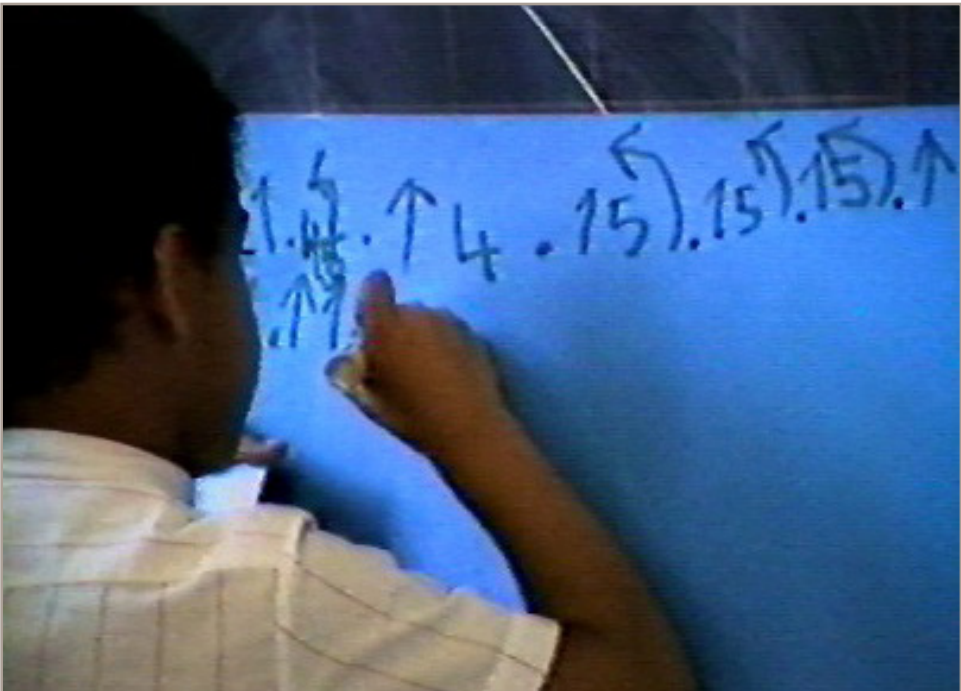


Figure 4 - Élaboration du pilotage d'une tortue Logo (GSM)

Crédit photo : Didier Vaudène.

la figure. Ce seul aspect nécessite d'avoir prévu un moyen simple, souple, une possibilité d'étalonnage pour le réglage informatique du nombre de pas-moteur pour la commande de chacune des roues. En effet, le tracé dépendra toujours de roues plus ou moins bien gonflées, de légères dérives ou glissements en pivotant, selon le matériau du sol. Et si la tortue d'écran n'est pas assujettie aux frottements et aux glissements des roues, elle subit aussi la finitude due à l'effectivité des calculs trigonométriques impliquant l'arrondi des résultats.

Les élèves de CP et de CE ont ainsi utilisé successivement des tortues de sol et des tortues d'écran. En fait, on peut considérer deux micromondes pour les tortues : celui du plan horizontal et celui du plan vertical. Leurs caractéristiques sont différentes quant aux repères et aux références à préciser (usage d'une grande feuille au sol ou de la cour en EPS ; la direction 0, une position d'origine, etc.), ou bien imposées pour la tortue d'écran. Les enfants découvrent ces deux mondes, avec quelques difficultés pour certains. Avec le recul, nous regrettons de ne pas avoir suffisamment étudié comment s'effectue ce passage du monde horizontal au monde vertical.

Une tortue tutorielle

Dès l'introduction, nous avons souligné le caractère quelque peu paradoxal que ce soit dans un contexte d'intelligence artificielle et de langage évolué que

le projet Logo se soit développé. Dans le cas de la tortue, pour le regard parfois distrait d'un adulte, qu'y a-t-il de plus simple à comprendre en son principe qu'un tel dispositif – un jouet, peut-être – conçu pour des enfants ? Or, dans l'expérience d'une pratique active de l'informatique, c'est d'abord pour elle-même que l'informatique intervient, comme une fin, ce qui n'empêche évidemment pas qu'elle intervienne aussi en tant que moyen – un outil, comme on dit – au service d'autres apprentissages. C'était déjà avoir pris acte de l'importance de pratiques, de schémas, de concepts et de raisonnements, venant à côté ou en complément des contenus traditionnels, aussi bien dès l'école maternelle que dans la vie courante, qui tendent maintenant à se réifier comme « *computational thinking* » (Wing, 2006), raison numérique ou raison computationnelle (Salanskis, 2011), pensée informatique (Berry, 2019), etc., même si parfois un enthousiasme excessif pour ces nouveautés produit des exagérations et des réductions abusives amalgamant tout cela dans une vulgate mécaniciste et programmatique digne du retour à un pythagorisme primitif antérieur à la catastrophe des irrationnels (auquel le syntagme « le numérique », avec son extension étonnante, apporte peut-être un très fervent soutien).

Pourtant, ce qui torture l'esprit, ce qui est peut-être pour lui le plus difficile à atteindre, c'est la simplicité, un noyau à la fois minimal et fondamental à partir duquel puisse être dépliée ou re-dépliée une extension maximale. Une telle simplicité, qui est tout autre chose

qu'un simplisme, peut cependant être mise en œuvre, dans certains cas, grâce à des dispositifs suffisamment simples pour intéresser les enfants. Nous voudrions un instant attirer l'attention sur quelques traits d'une telle simplicité qui sont plus particulièrement mobilisés dans une pratique active de l'informatique, dès la grande section de maternelle, et en esquisser brièvement quelques enjeux et incidences (qui peuvent aussi intéresser les grandes personnes).

Pour une « cyber-dialectique »

Peut-être l'un des apports remarquables de ces expériences (et aussi, à bien des égards, de l'informatique elle-même), est l'attention portée à l'analyse des écarts

entre le projet et la réalisation, écarts nommés ordinairement des « erreurs », qui se manifestent en particulier, dans le contexte de Logo, par les écarts entre les dessins que les enfants imaginent, projettent et dessinent, et les tracés qu'ils obtiennent quand ils pilotent ou programment la tortue. C'est en effet tout ce cycle réflexif qui est un puissant stimulant, y compris dans des élaborations collectives où chacun peut autant contribuer au projet qu'à sa réalisation et à son éventuelle critique et correction. La formalité et la passivité (au sens où il n'y a pas d'interventions intentionnelles) des dispositifs informatiques utilisés signifient que rien ne se produit qui n'ait été effectivement commandé ou programmé, de sorte que ces dispositifs agissent comme des miroirs qui permettent à chacun de recher-



Figure 5 - Débogage graphique grâce à une « correction manuelle de trajectoire » (GSM)

Crédit photo : Didier Vaudène.

cher l'ajustement entre ce qu'il souhaite ou imagine et ce qu'il obtient effectivement (figure 5). Il peut ainsi insensiblement, et peut-être sans même en prendre conscience, accepter et intégrer cette « culture de l'erreur » qui vise à traquer ces difficultés (et non à les fuir, car elles sont l'occasion d'une avancée), à les attendre (et non à les craindre, car elles sont [presque] inévitables), afin de les interpréter, de les comprendre, de les corriger, et d'éviter de les reproduire, tout cela sans l'intervention parfois dramatique d'une parole magistrale prononçant une sanction négative ou un jugement défavorable.

Dans le petit groupe de quelques enfants d'un projet, c'est la dynamique propre du groupe qui offre à chacun la possibilité de jouer tous les rôles, quand il contribue à l'analyse et à la conception du projet, quand il discute une décision et cherche à se la faire expliquer, quand il défend une option et tente de convaincre ses camarades de l'adopter, ou quand il cherche à interpréter le hiatus entre le projet imaginé et l'effectuation qui en résulte. Aussi une telle dynamique évoque-t-elle aussi bien un pilotage cybernétique, dans lequel un système observe et contrôle son propre fonctionnement, qu'un dispositif expérimental où la passivité du robot figure une stabilité (au moins relative) des conditions d'expérimentation. Mais elle met aussi en jeu ce qui peut être dialectiquement (au sens fort de l'intervention d'un dialogue) et inter-subjectivement élaboré, argumenté et construit : dans une telle « micro-république de savoir », la loi du plus fort n'est pas la meilleure (figure 6).

Aussi une telle pratique active favorise-t-elle le développement de raisonnements dont le nombre d'étapes et la complexité vont croître progressivement, et cela bien avant que l'exigence de la démonstration – qui ne s'acquiert que plus tardivement, au niveau du secondaire, dans le contexte des mathématiques – ne soit abordée, thématisée et mise en œuvre comme telle.

De l'« écriture » avant l'écriture

Certes, cette dynamique d'élaboration comme pratique active peut être mise en jeu dans des contextes divers. Toutefois, l'informatique joue un rôle particulier en tant qu'elle intéresse directement l'articulation entre l'empirie des choses et ce qui se laisse approcher par la médiation de l'écriture, aussi bien quand il s'agit de saisir quelque chose comme de l'écriture, que quand il s'agit d'appliquer, à un dispositif ou à un contexte empirique, ce qui a été conçu ou compris depuis ou via l'écriture. D'où un effet potentiel d'attraction très puissant entre l'informatique et les disciplines ou les domaines qui sont étroitement liés à l'écriture et à sa formalité. C'est en particulier le cas pour les mathématiques et, surtout, la calculabilité, les nombres et jusqu'aux emblématiques « 0 » et « 1 » de l'informaticien. Mais, tout comme la machine d'arithmétique de Pascal, réalisée à la façon des horlogers, ces machines de traitement de l'information sont des machines sans écriture, et ce n'est que sous le couvert d'une fiction qu'on peut affirmer que « tout » (c'est-à-dire pas tout) se passe comme s'il s'agis-



Figure 6 - Une étape de la réalisation d'un projet Logo en petit groupe sur un Micral 8022G (CE1)

Crédit photo : Didier Vaudène.

sait d'écritures et d'opérations appliquées à des écritures²¹ (Vaudène, 2021). D'où une élasticité d'usage de ces machines qui est bien plus ouverte qu'on ne le croit parfois, et c'est une telle élasticité qui est en jeu aussi bien dans l'élaboration de Logo que dans cette expérience d'une pratique active de l'informatique.

²¹ En toute rigueur, ce « comme si » signifie que ce n'est pas à proprement parler la médiation de l'écriture qui intervient, mais la médiation de l'information (non pas l'information probabiliste au sens de Shannon, mais l'information discrète au sens de l'informaticien), médiation qui est plus fondamentale et qui « porte » en quelque manière la médiation de l'écriture.

L'activité d'éveil introduite en grande section de maternelle autour d'une tortue Logo permet déjà une mise en œuvre d'opérations et d'enchaînements par des enfants qui ne savent pas encore lire ni écrire, mais qui commencent à reconnaître les chiffres et les lettres. On comprend que la fiction du « comme si » préserve une distance à l'égard de la formalité écrite (au sens de l'écriture graphique ordinaire), et de même que les machines de traitement de l'information sont des machines sans écriture, de même peut-on effectuer ou faire effectuer des calculs qui n'auront pas été produits ni effectués dans le cadre

d'une formalité écrite. C'est aussi une manière de rappeler qu'il y a aussi une distance entre calcul et nombre, et que l'association entre calcul et nombre n'est qu'un cas particulier – pour historique qu'il soit peut-être – et que les calculs qu'on effectue, de quelque manière qu'on s'y prenne, ne concernent jamais des nombres en tant que tels – ce sont des idéalités – mais des lettres (ou ce qui en tient lieu sous le couvert de la fiction du comme si). Ce que Ludwig Wittgenstein résume synthétiquement : « *Le nombre n'est absolument pas un "concept mathématique fondamental". Il y a tellement de calculs dans lesquels il n'est pas question de nombres* » (Wittgenstein, 1969, p. 297, passage cité dans Bouveresse, 1980, p. 25). C'est déjà cette double distance qui permettait, par exemple, à un Pascal de construire sa machine, à un Leibniz de concevoir l'idée de la Caractéristique, etc., et, plus récemment, aux machines de traitement de l'information de voir le jour. Wallace Feurzeig, l'un des concepteurs du langage Logo, résume très bien l'impulsion initiale de Logo (Walden, 2011, p. 290) :

Mes collaborateurs dans cette recherche furent Daniel Bobrow, Richard Grant et Cynthia Solomon de la BBN [Bolt, Beranek & Newman] et un consultant Seymour Papert, qui était arrivé récemment au MIT depuis l'Institut Jean Piaget de Genève. L'idée d'un langage de programmation expressément conçu pour les enfants émergeait directement du projet TELCOMP²².

Nous avons réalisé que la plupart des langages existants étaient conçus pour faire des calculs et qu'ils ne facilitaient généralement pas la manipulation de symboles non numériques. Les langages de programmation ordinaires étaient inappropriés pour un usage éducatif : on les utilisait souvent dans des expressions écrites de type extensif qui n'allaient pas dans le sens du caractère expressif des étudiants ; ils avaient de sérieuses déficiences dans les structures de contrôle ; leurs programmes manquaient de constructions procédurales ; la plupart n'avaient aucune disposition qui puisse faciliter les définitions dynamiques et leur exécution ; quelques-uns, rares, disposaient de moyens pour le débistage des pannes, pour le diagnostic et l'édition des programmes, si indispensable aux applications éducatives.

Le besoin d'un nouveau langage conçu et dédié à l'éducation était évident. Les bases requises pour ce langage étaient que :

1. de jeunes élèves doivent être capables de l'utiliser pour des travaux éducatifs élémentaires avec très peu de préparation ;
2. sa structure doit englober des concepts mathématiques importants avec un recours minimal aux conventions de programmation ;
3. il doit permettre aussi bien l'expression d'algorithmes élaborés non numériques que l'expression d'algorithmes numériques.

Avec les premières tortues de sol, en 1971, c'est une nouvelle distance qui

²² Langage de programmation conversationnel et interprété, que BBN venait d'implémenter en 1964, et

dont la syntaxe est proche de celle de BASIC, elle-même dérivée de FORTRAN.

intervient, puisque c'est l'ordinateur lui-même, dans l'une de ses configurations typiques – clavier, processeur, écran – qui est ainsi tenu à l'écart, ce qui provoque une redistribution des rôles. C'est d'abord la programmation vécue, évoquée précédemment, qui fait jouer la fiction du comme si, pourrait-on dire, en sens inverse : il n'y a certes aucun programme qui soit enregistré comme tel, mais il y a quand même programmation. Dans un tel dispositif, l'implémentation matérielle de la connexion entre l'introduction des cartes dans le lecteur et les déplacements de la tortue est certes mise entre parenthèses, mais elle ne disparaît pas pour autant, faute de quoi le dispositif ne fonctionnerait pas : l'effectivité de la connexion est en quelque manière recueillie au plan cognitif par les enfants à la manière d'un verbe d'action : quand j'insère cette carte dans le lecteur, la tortue fait ceci. Et c'est le signifiant de ce verbe qui sera ultérieurement récupéré pour donner lieu à l'inscription, au nom, au symbole, à l'icône, etc., de cette opération.

Rien n'impose alors que cette mise entre parenthèses soit nécessairement prise en charge par le dispositif matériel « tortue Logo », et rien n'empêche que la connexion entre la commande et l'action puisse être prise en charge... par les enfants eux-mêmes ! C'est ce qu'a très vite aperçu la professeure d'éducation physique et sportive, Madame Charbonnier, en inventant divers exercices inspirés de Logo pour les CP et les CE. Par exemple, un élève qui a les yeux bandés doit franchir une zone parsemée d'obs-

tacles en étant piloté, en « commande vocale », par des commandes Logo que ses camarades lui donnent (figure 7).

Non seulement il s'agit encore d'une programmation vécue, mais c'est maintenant aussi un élève qui interprète les commandes données verbalement et qui, avec ses propres déplacements, prend complètement en charge la connexion entre la commande et l'action qui doit lui correspondre. On peut ainsi programmer durant les séances d'éducation physique et sportive, et le même principe peut s'appliquer à de nombreux exercices intéressant la confiance en autrui, les déplacements, la latéralisation, l'orientation et les changements de repères (ma gauche, ma droite, celles du voisin à côté de moi, à ma gauche, à ma droite, devant moi dans le même sens, face à moi, la gauche et la droite de la tortue ou de son substitut, etc.). Comme on le voit sur ce photogramme (fig. 7), personne n'a de papier ni de crayon, et il semble que l'écriture soit complètement absente. Elle est pourtant bien présente partout sous le couvert du « comme si » inversé, et chacun peut ainsi toucher du doigt le moment où le langage vient en quelque manière tangenter l'écriture. Et ce n'est donc pas seulement non-numérique, ni autre chose que la manipulation de symboles, c'est en participation avec les symboles, mieux, c'est de la symbolisation s'accomplissant, incorporée, comme une « écriture » qui viendrait « avant » l'écriture²³.

23 Cette indication d'une « écriture » qui viendrait « avant » l'écriture recroise aussi bien la problématique



Figure 7 - Un élève, les yeux bandés, piloté en « commande vocale »

Crédit photo : Didier Vaudène.

Jouer le rôle de l'interprète

Nous avons évité avec soin d'approcher ces exercices d'EPS en recourant à l'expression métaphorique ordinaire de l'« enfant-robot » – jouer au jeu de l'enfant-robot, comme on le dit dans les articles concernés par Logo – pour lui substituer l'idée d'interprète, autrement plus féconde : l'enfant assume la fonction d'un interprète effectif, et ce sont en

déjà mentionnée du « comme si » et de l'information au sens de l'informaticien, que la question de l'archi-écriture abordée par Jacques Derrida dans *De la grammatologie* (Derrida, 1967, p. 83 et *passim*), y compris dans son rapport à l'information.

retour nos machines (tortues, ordinateurs, etc.) qui prennent rang de machines d'interprétation. On recroise ainsi très directement, d'une part, la problématique de l'effectivité qui concerne autant les dispositifs empiriques (processeurs, ordinateurs, etc.) que formels (démonstrations logiques et machines mathématiques, par exemple), et, d'autre part, les problématiques relatives à l'usage des systèmes ouverts ou à l'intelligence artificielle (ce que nous avons appelé la « programmation vécue »).

Ce que les exercices d'EPS mettent en jeu peut se comprendre comme une

substitution d'effectivités. Dès lors que la connexion entre une commande et son effet attendu est déterminée, on peut mettre cette connexion entre parenthèses, c'est-à-dire permettre la substitution des implémentations de cette connexion les unes aux autres, pourvu que cette détermination soit conservée. Autrement dit, ce qui importe, c'est que la connexion s'accomplisse réellement, qu'elle produise un effet conforme à sa détermination, sans qu'on ait pour autant à connaître comment s'effectue cet accomplissement. Dans le contexte d'une pratique active de l'informatique, cette substituabilité permet à chacun de jouer tous les rôles, y compris celui de la tortue ou de l'ordinateur, plus précisément, celui de l'interprète qui effectue la connexion entre les commandes (verbes, opérations, etc.) et les actions (effectivité). Cette substituabilité a une portée très générale, non seulement à l'égard des dispositifs scientifiques et techniques (Vaudène, 2016), mais aussi dans l'élaboration d'accords intersubjectifs (Vaudène, 2019). À cet égard, on pourra rappeler que la « machine » de Turing, en tant que machine mathématique, n'est pas une machine au sens de l'informaticien car il lui manque l'effectivité – la carte n'est pas le territoire –, du moins jusqu'à ce que le mathématicien prenne en charge cette effectivité, et ainsi la lui procure, en appliquant lui-même les règles d'interprétation, tout comme s'il jouait au jeu de l'enfant-robot (qui serait surpris qu'on dise qu'une telle « machine » est, quant à son « implémentation », une machine sans écriture ?). Dans son article de 1936,

Turing précise : « *Un homme en train de calculer la valeur d'un nombre réel peut être comparé à une machine susceptible de se trouver dans un nombre fini d'états [...]* » (Turing, 1936, p. 51). Dans ce texte, ce qui calcule, c'est un homme, de sorte que ce devrait être à la machine de jouer au jeu de « l'homme-calculant » pour imiter un homme en train de calculer (en 1936, les ordinateurs n'existent pas, de sorte que Turing ne peut pas faire référence à nos machines informatiques pour inverser les rôles). Dans sa thèse, Patrice Pissavin résume synthétiquement l'enjeu de cette substituabilité (Pissavin, 2019, p. 166) :

1. la machine de Turing est une expression de l'exigence générale d'intersubjectivité d'une démonstration en mathématique dans le contexte de la vérification, lorsque celle-là est appliquée à un système formel ;
2. l'exigence générale d'intersubjectivité d'une démonstration en mathématique, implique la nécessité que cette démonstration soit publiquement communicable (en un temps fini) et que sa vérification ne fasse appel à aucun engagement invérifiable en principe par n'importe quel être humain.

On peut déjà comprendre que le mécanisme prêté aux machines n'est que l'une des facettes d'un réseau d'articulations qui comprend aussi ce qu'on peut entendre comme une exigence de neutralité dans la constitution des accords intersubjectifs (et cette ascèse, aussi astreignante que corrosive, n'est pas sans rappeler, quant à sa radicalité, celle que

Platon prête à Socrate à l'égard de ses interlocuteurs). Mais on peut aussi comprendre, ce qui ne va pas tout à fait de soi, que c'est seulement faute de mieux ou d'autre chose qu'un consensus se résigne à recourir à de tels dispositifs (machines de Turing ou équivalents) pour caractériser cette exigence de neutralité, et ainsi préciser ce qu'implique l'exigence d'effectivité des démonstrations. Et si les théorèmes négatifs obtenus, par exemple, dans les théories de la démonstration ou de la calculabilité, dépendent de cette exigence, alors il faut aussi comprendre que l'effectivité empirique de ces dispositifs intervient comme une condition de la possibilité de ces théories, en même temps qu'elle en constitue une limitation énigmatique – comme une manière de point aveugle –, désignant ainsi le lieu d'une ouverture²⁴. Jouer au jeu de l'enfant « robot », c'est jouer le rôle de l'interprète, et c'est donc aussi prendre en charge ce point aveugle comme un accomplissement vécu.

Sans doute la première étape de l'utilisation de la tortue Logo en mode « programmation vécue » peut-elle paraître très rudimentaire, et certes elle l'est quand on attend les étapes suivantes qui introduiront le schéma de contrôle avec ses alternatives, ses itérations, ses procédures, ses fonctions, ses récursions, etc., et, surtout, l'enregistrement du programme. Mais cette programmation à la

²⁴ Concernant l'articulation entre la constitution des consensus, les limitations et points aveugles, l'effectivité et l'ouverture, voir (Vaudène 2017).

volée n'est pas une étape archaïque de la programmation, qu'il faudrait dépasser, car elle appartient à l'ouverture, c'est-à-dire au fait que le système informatique concerné, en l'occurrence la tortue, agit, en tant que machine d'interprétation, comme un système ouvert. En d'autres termes, si le principe du programme enregistré s'accorde à la figure de la fonction mathématique calculable, l'un et l'autre s'accordent aussi à un usage ou à un mode de fonctionnement clos, requérant un programme élaboré, énoncé et enregistré avant son application, c'est-à-dire en somme un processus laplacien supposé se dérouler sans aucune intervention extérieure : l'idée d'entrée-sortie demeure étrangère au concept mathématique de fonction. Mais, dans le cas d'un dispositif ouvert, certaines séquences d'interventions extérieures au dispositif peuvent jouer le rôle d'un programme qu'on énonce et qu'on introduit à la volée.

Apprentissages et expériences en cours moyen

Les mots pour le dire

Sur les différentes facettes de la pratique active de Logo se sont greffées, au cours des années suivantes, des recherches sur les capacités des élèves, en CM principalement²⁵, à élaborer leur propre dictionnaire d'identificateurs pour

²⁵ Classes de mesdames Benisty et Viguier.

les fonctions standard du langage Logo, sachant que des systèmes-langage comme Lisp ou Logo peuvent parfaitement s'adapter à de tels dictionnaires personnalisés par simple effet de redéfinition.

Un exemple permet de situer rapidement l'idée : Logo possède des fonctions de traitement de chaînes de caractères permettant d'opérer sur des caractères individuels, des mots et des phrases. La fonction qui permet d'extraire le

premier caractère d'un mot (mémorisé dans une chaîne) se nomme, suivant les versions : P ou PREM ou PREMIER. Les formes abrégées sont hélas d'autant plus absconses qu'elles sont plus concises, tandis que la variabilité des noms selon les versions de Logo ne facilite guère le passage d'un système à un autre, surtout pour de très jeunes informaticiens. Une classe a rapidement produit et accepté son identificateur : `Le_premier_caractère_du_mot` (Nowak & Le Roch, 1984).

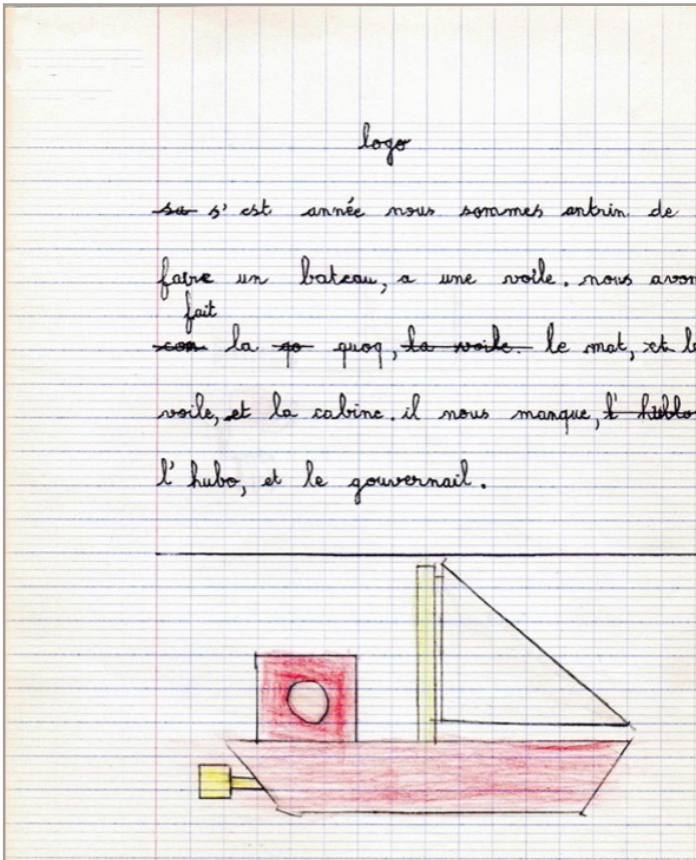


Figure 8 - L'amorce d'une programmation modulaire (CE1)

Crédit photo : Gérard Nowak.

L'exercice est-il anodin ou seulement une cosmétique superficielle ? On recroise ce qui a été déjà suggéré pour l'articulation entre le verbe d'une action et le nom d'une commande : choisir avec soin l'identificateur d'une fonction, c'est d'abord en explorer la sémantique et la comprendre, et c'est aussi en faire une synthèse. La leçon vaut au cours moyen comme elle vaut à l'université, et *L'Art poétique* de Boileau, à cet égard, n'a rien perdu de son actualité, surtout quand on transpose certaines de ses recommandations à cette poétique particulière qu'est la programmation : ce que l'on conçoit bien s'énonce (et se programme) clairement, et l'identificateur pour le nommer arrive aisément. Un choix judicieux d'identificateurs est loin d'être un exercice simple, car il doit satisfaire à diverses contraintes pour qu'une bonne cohérence du dictionnaire auquel il contribue puisse être obtenue : contraintes syntaxiques, morphologiques, de longueur (ni trop long, ni trop court), filtrage des incohérences de langue (éviter le mélange des langues, de trop écorcher les mots, par exemple), et favoriser la qualité mnémotechnique en gardant trace le plus possible des traits essentiels de la sémantique de la fonction ainsi nommée (figure 8). Et quand l'exercice concerne une classe, c'est encore une pratique active qui conduit à un travail collectif et coopératif où une « obligation de résultat » est attendue, ce qui suppose de savoir manier les justifications et les arguments, tout en sachant accepter les compromis car les possibilités de choix sont multiples.

Dans cette perspective, les termes d'un langage de programmation ne s'imposent plus de manière incontournable ; on peut refuser d'utiliser les identificateurs fournis quand ils sont jugés inappropriés, trop abrégés, franglais, etc. Et quand les élèves ressentent le besoin de produire des abréviations, ce sont encore leurs propres choix créatifs qui sont mobilisés, à l'issue d'un processus qui a commencé par déplier et formuler la sémantique de la fonction.

Répétition

Prendre de la distance par rapport à une programmation complètement séquentielle à la volée, dans laquelle les commandes sont introduites et exécutées une à une, revient à introduire les structures de contrôle élémentaires, au moins l'alternative, l'itération et la procédure²⁶, ce qui implique que tout ou partie du programme soit enregistré, au moins les séquences placées sous le contrôle de ces structures. Ce qui implique le recours à l'écriture, éventuellement de manière indirecte via des substituts graphiques ou matériels qui sont *in fine* mis en œuvre comme s'il s'agissait d'écritures.

Dans des classes de CM²⁷, dont aucun élève n'avait eu de formation à l'informatique les années précédentes, on s'est intéressé à ne pas « parachuter » les

²⁶ D'autres choix sont possibles.

²⁷ Classes de Madame Benisty en CM1, et de Messieurs Calladine et Schroeder en CM2.

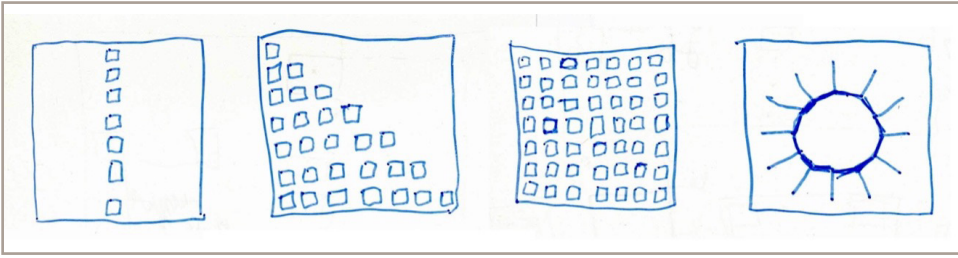


Figure 9 - Énoncés graphiques d'exercices

Crédit photo : Gérard Nowak.

structures de contrôle, et à étudier d'abord la capacité des élèves à reconnaître des motifs graphiques simples présentés sur papier et comportant des répétitions, à tenter de les nommer, de formuler, dans des termes qui leurs soient propres, la reconstruction de ces motifs à l'écran, et à dégager le test ou la borne permettant d'arrêter la répétition (figure 9).

Les énoncés des élèves ont pris diverses formes, par exemple : « fais une colonne de 7 carrés de 1 cm de côté et séparés de 0,5 cm », « fais 28 carrés de 1 cm séparés de 0,5 cm en escalier », « remplis l'écran de carrés de 1 cm de côté », « fais un soleil en 12 segments et ayant 12 rayons », etc.

Il n'en reste pas moins que les difficultés sont nombreuses, même si la traduction graphique paraît aller de soi, car le motif de la répétition ne se dégage bien souvent que lorsqu'on dégage correctement la variabilité corrélatrice d'un facteur qui peut être une position, une orientation, etc. Au moins dans les cas simples, la première réaction est le plus souvent celle de l'énoncé séquentiel exhaustif sans

qu'un schéma de répétition ne soit dégagé comme tel (il ne faut pas non plus sous-estimer, surtout à cette époque, que le plaisir des élèves à taper au clavier et à monopoliser un ordinateur était plutôt un obstacle à la recherche des motifs répétitifs conduisant à des programmes plus concis !).

Avant que ne soient introduites les structures de contrôle d'un langage, ce qui s'accompagne d'une exigence de précision et d'exhaustion des difficultés à résoudre, ce que les élèves produisent s'apparente plus à des descriptions, plus ou moins détaillées ou précises, qu'à des embryons d'algorithmes (figure 10).

On observe en particulier de nombreuses erreurs et imprécisions pour les actions à effectuer entre deux motifs graphiques répétés. Parmi les expressions qui reviennent fréquemment, on peut mentionner quelques formes typiques pour exprimer la répétition concernant la séquence d'instructions dessinant un motif :

- <séquence> x <un nombre> (où le « x » est à lire : multiplié par),

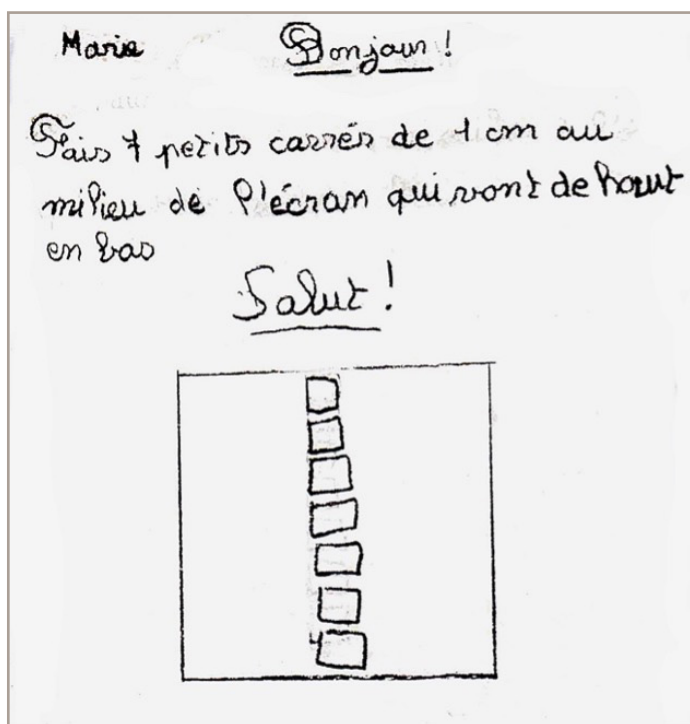


Figure 10 - Esquisse d'une répétition d'un motif

Crédit photo : Gérard Nowak.

- <séquence> <un nombre> fois ;
- <séquence> etc. ;
- <séquence> pendant <un nombre> fois ;
- <séquence> et recommence <un nombre> fois ;
- <séquence> fais <un nombre> fois pareil ;
- <séquence> jusqu'à ce que <condition d'arrêt> (rares occurrences)

Il y a de rares propositions de répétition avec un changement de valeur : « fais <séquence> comme pour l'autre

dessin, sauf que ce n'est pas le même nombre », et encore plus rarement des propositions de répétitions emboîtées comme « 7 rangées de carrés... avec 7 carrés dans une rangée ». Dans la proposition ci-dessous, on peut observer que le carré individuel est présenté in extenso, et que la répétition ne concerne que le fait de répéter 6 carrés identiques décalés de 10 pas (figure 11).

Dans les activités qui ont été effectuées, le schéma d'une répétition comportant un compteur modifié à chaque itération n'a jamais été rencontré. De même, nous avons pu constater que les

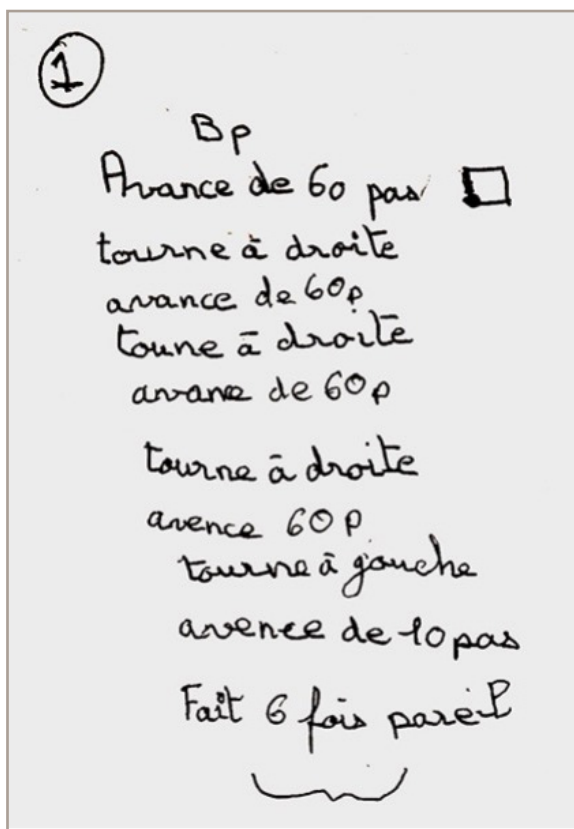


Figure 11 - Proposition pour une répétition avec motif

Crédit photo : Gérard Nowak.

élèves ne transposent pas intuitivement des schémas du quotidien pourtant déjà bien connus – une liste (d’objets à emporter), une pile (de livres), une rangée (de livres sur une étagère), un ensemble d’étagères (de livres) superposées, un annuaire (de téléphone), un tiroir (de commode), etc. – pour être utilisés – liste, pile, tableau, table, dictionnaire, variable, etc. – dans un contexte informatique. Diverses expérimentations ultérieures ont également mis en évidence des difficultés analogues de transposition concernant l’articulation

entre des activités Logo et des concepts ou des problèmes mathématiques, aussi bien géométriques qu’algébriques (Clements & Meredith, 1992).

Ce type d’activités avait aussi été proposé et accepté au sein d’une classe de perfectionnement de niveau CE2-CM1 (classe de Madame Salvaing), pour une dizaine de séances, les samedis matin. L’enseignante faisait remarquer que l’élève souvent le plus distrait ou le plus dissipé se montrait là le plus participatif

et productif. Dans ce cas, les exercices avaient concerné des opérateurs sur les chaînes de caractères (par exemple : extraire le *n*ème caractère d'une chaîne). Les notions de « boîte noire », avec nom de l'action, entrées de données, et sortie d'un résultat, et de « diagramme de Conway » ont bien aidé le travail et ont été plutôt bien utilisées par des élèves.

L'intérêt de telles expériences est en particulier d'attirer l'attention sur ce qui ne va pas de soi pour les élèves, et qui devra donc être l'objet d'un apprentissage de la part des élèves et d'une attention particulière des enseignants. Il est en effet déjà souvent difficile de reconstituer après-coup, une fois un principe compris, quelles furent les difficultés et les obstacles qu'on a dû surmonter pour y parvenir ; mais c'est encore plus difficile quand ce sont de jeunes élèves qui sont concernés, et que les notions à introduire, quand elles sont nouvelles, ne sont pas accompagnées d'une expérience pédagogique bien établie.

Dans les années 1980, avec le développement de langages fonctionnels récurifs comme Lisp et Scheme, et leur utilisation pour l'apprentissage de la programmation dans l'enseignement supérieur, des chercheurs américains (et même français) ont tenté de faire valoir que la notion de récursivité pouvait être considérée comme « naturelle » même pour des enfants (de 9 à 11 ans). Dans le contexte de Logo et des expériences que nous avons menées, nous n'avons trouvé qu'une seule proposition (sur une

cinquantaine) qui soit apparentée à une forme récursive (à moins que cette hypothèse osée ne concerne que des enfants de chercheurs, voire de ceux-là seulement).

Trier, comme un jeu

Voici quelques aspects d'une activité menée en CM²⁸ sans préparation particulière au problème des tris, sans usage d'ordinateur, et auprès d'élèves qui n'ont pas encore été influencés par un langage informatique. Sur une demi-journée, il s'agissait de présenter une « consigne », de laisser le temps de la réflexion et de l'élaboration de solutions individuelles, puis de collecter les résultats, et les analyser ; et enfin, lors d'une seconde session, de demander à certains élèves de préciser leur solution, éventuellement en leur demandant de « jouer » les actions écrites pour leur faire apercevoir une imprécision ou une difficulté. Il s'agit donc d'un travail de conception et d'imagination, sans intervention d'ordinateur, dans lequel les élèves doivent aussi prendre en charge le contrôle de ce qu'ils proposent. On peut souligner que, bien que les problèmes de tri soient un peu délicats à appréhender, des propositions d'élèves se sont approchées d'une solution correcte.

La « consigne de tri » est formulée comme une mise en scène impliquant des actions de communication et de déplacement, ce qui permet ainsi aux

²⁸ Classes de Madame Viguier en CM1 et de Monsieur Calladine en CM2.

élèves d'imaginer aisément le cadre du problème, et aussi, le cas échéant, de représenter leur solution comme un jeu ou comme une sorte de pièce de théâtre :

Consigne de tri. Vous êtes 20 dans la classe. Je vous transmets secrètement à chacun un numéro différent compris entre 1 et 100. Comment communiquez-vous pour vous coordonner et vous retrouver classés, d'après votre numéro, le long du mur du tableau ?

Les déclinaisons intuitives des élèves sont d'autant plus remarquables, qu'elles sont dignes d'algorithmes de tris tout en correspondant à des embryons de méthodes, qui, elles, étaient enseignées à l'époque en 1^{re} année du cycle supérieur. Mentionnons quelques propositions :

- Division d'un problème en sous problèmes (Sani) : « Ceux qui ont un numéro entre 0 et 20 vont se mettre quelque part, après eux, ceux qui ont un numéro entre 20 et 40, et entre 40 et 60... »
- Tri de type « *bubble sort* » ou « min-max » (Christophe) : « Pour aller en ordre croissant : on prend le plus petit et le plus grand des nombres, on les classe, puis on prend encore le plus grand et le plus petit des nombres restants jusqu'à ce qu'il ne reste plus un seul nombre. Si à la fin il ne reste plus qu'un chiffre, on le met au milieu. Après on appelle les élèves qui ont les numéros inscrits. »
- Tri dichotomique (Jérôme) : « On demande à chacun son numéro, s'il est plus petit que 50 on le met d'un

côté s'il est plus grand que 50, on le met de l'autre côté et on recommence avec 25 et 75. »

Sur ces quelques réponses de grand intérêt, on peut prendre le temps d'essayer d'affiner le cas d'indécision pour les élèves ayant les numéros 20, 40, 60, 80, cités deux fois (Sani) et de questionner sur une suite du procédé après « entre 60 et 80 » et « entre 80 et 100 ». Idem pour la méthode dichotomique (Jérôme). Il faut – du moins il faudrait – aussi examiner – mais avec beaucoup plus de séances – les réponses moins remarquables, parvenir à rendre explicites certaines propositions, et admettre que ces activités sont sans réponse pour certains, parce que la consigne n'a pas été comprise, ou dont la réponse se réduit à un exemple qui se borne à reformuler la consigne, ou qui paraît hors sujet. Par exemple, Olivier : « Nous pouvons nous ranger par télépathie, par signes alphabétiques », ou François : « Par télépathie, de rangée en rangée ». Après quelques demandes de précisions, il s'avéra que cette « télépathie » était en fait un dialogue sans paroles, au moyen de signes avec les doigts pour se communiquer les numéros... ce qui relance l'intérêt de la réponse. C'est un exemple qui signifie qu'il aurait fallu présenter aux élèves quelques opérateurs primitifs, construits sur un schéma simple d'envoi et de réception de messages, permettant de faciliter la réflexion des élèves dans un contexte qui met explicitement en jeu de la communication et de la coordination (Darche & Nowak, 1992).

Communiquer pour se coordonner

Nous venons de souligner l'aspect algorithmique de la consigne proposée. Mais cette consigne, dans sa concision destinée à laisser très ouvert le champ des solutions possibles, comporte un autre volet orienté vers la communication entre les élèves quand ils essaient de réaliser leurs propositions de solutions en se déplaçant dans la classe. Revenons sur la consigne de tri pour souligner cet aspect de communication qui s'ouvre à son tour sur une relative indépendance des élèves : rien n'impose en effet, dans la réalisation d'un tri donné, qu'un seul élève soit actif à la fois, de sorte que la consigne peut tout à fait conduire à des solutions parallélisées de ce problème de tri. On peut aisément deviner qu'une telle activité puisse conduire à de nouveaux problèmes de coordination et de synchronisation, de réservation et de libération de ressources, d'attentes éventuelles, etc.

Ce que les élèves peuvent ainsi commencer à découvrir par leurs propres déplacements peut aussi être mis en scène par d'autres moyens. Prenons l'exemple d'une ressource à accès exclusif. On peut imaginer l'utilisation de deux tortues de sol, disposées sur une ligne de départ et devant atteindre une ligne d'arrivée parallèle à la ligne de départ. Dans le parcours, on dispose un pont à voie unique. Très certainement, chaque tortue sera programmée pour « foncer » et arriver la première au pont. Ce n'est pas la solution. Le pont est une ressource à accès exclusif,

qui doit être réservée, puis utilisée quand on obtient l'autorisation, et enfin libérée pour laisser l'accès à une autre tortue qui aura peut-être été mise en attente.

Si ce sont des enseignants, autant du primaire que du supérieur, qui ont d'abord nourri ces expérimentations concernant le parallélisme, ce sont ensuite des solutions remarquables, produites par les enfants dans le cadre de ces expérimentations – et aussi d'autres sujets, comme la résolution de puzzles à l'aide d'un anneau d'acteurs communicants par messages –, qui ont fait l'objet de sujets de projets au niveau Maitrise en utilisant le système Unix, ses processus, ses sémaphores, etc., tout cela en vue de la création d'outils pédagogiques d'aide à ces apprentissages avec livraison d'applications réutilisables.

Conclusion et prolongements

L'expérience suivie sur six ans dans l'école était nouvelle. Malgré les difficultés que nous avons rencontrées, nous souhaitons que l'expérimentation puisse motiver toutes les classes d'une école et leurs enseignants et, au sein d'une classe, tous les élèves. L'expérience a suscité l'intérêt de toutes les parties : parents, enseignantes, directeurs d'écoles, inspecteurs, mairie du 15^e, quelques collègues pionniers... et surtout celui des enfants ! Cependant, tout en demeurant ambitieux et enthousiastes pour des projets ultérieurs, il nous fallait rester modestes. Il

faut prévoir, au sein d'une classe, le désintérêt complet de quelques élèves, qui survient parfois après quelques échecs, comme certains en ont témoigné dans leurs textes de fin d'année. L'une des causes – parmi bien d'autres – de ces désintéréts réside à notre sens dans le fait que les langages proposés ont aussi pour effet d'imposer des choix lexicaux et syntaxiques, souvent sans aucune adaptation possible, alors que la possibilité d'une adaptation est une richesse. Il faut aussi accepter qu'au sein de l'école – 5 niveaux dédoublés, soit 10 classes –, et après six années d'efforts, nous ne soyons parvenus à motiver qu'un peu plus de la moitié des enseignants. Est-il nécessaire d'ajouter que nous aurions souhaité une meilleure dotation en matériel ? Et surtout un soutien effectif en ce qui concerne la formation, complètement prise en charge bénévolement tout au long des six années de l'expérience.

Les multiples facettes de cette expérience, poursuivie dans la durée, ont produit une synergie qui a été une source importante pour de nombreux sujets de recherche pour des étudiants de DEA (Diplôme d'études approfondies) ou de DESS (Diplôme d'enseignement supérieur spécialisé), ou en thèse, et pour des publications, y compris d'autres vidéogrammes, par exemple (Vaudène & Nowak, 1985). De nouveaux outils pédagogiques, des applications, des systèmes-langages ont vu le jour au sein de notre groupe (Nowak & Le Roch, 1989, 1990). Des travaux de recherche ont en outre concerné d'autres micromondes

qui devaient permettre de faire évoluer des langages tels Lisp, Logo, Smalltalk, et même les langages pour acteurs. À la demande du MEN, nous avons participé, à l'UPMC, à la création de formations, parmi les premières dans ce domaine pour un contexte institutionnel, qui ont accueilli pendant deux ans des sessions de professeurs stagiaires des ENI et de futurs inspecteurs. Dans un autre registre, la compétence acquise grâce à cette expérience a été mobilisée pour des expertises ou des contributions, destinées au MEN, concernant des fournisseurs de matériels (société Jeulin) et de langages à vocation pédagogique (Logo Éducation nationale).

L'expérience a cessé quand elle a « rencontré » la grande vague de l'équipement des écoles via l'opération Informatique pour tous, ce qui s'est rapidement révélé correspondre à d'autres objectifs, donc à un autre monde (ce qui mériterait une autre étude). Le gouvernement souhaitait motiver non seulement, entre autres, tous les enseignants de toutes les écoles de France par un apport massif de formation et de matériel, mais en fait toute une société. Par humour ou peut-être surtout par goût de la provocation, on a pu entendre le professeur Marcel-Paul Schützenberger s'exclamer, dans un amphithéâtre de sensibilisation à l'informatisation de la société, dans le contexte du rapport *L'Éducation et l'informatisation de la société* (Simon, 1981), que c'était une bonne chose décidée, car au moins les élèves apprendraient à taper à la machine ! Mais rien ne va sans difficultés, et rien n'a été simple ni pleinement



Figure 12 - Programmation coopérative (CE1)

Crédit photo : Didier Vaudène.

réussi, ni pour notre expérimentation, à son échelle modeste, ni pour ce plan de grande envergure.

Malgré ces quelques réserves, le bilan global de l'expérience nous a paru – et nous paraît encore – largement positif. Elle a été humainement très riche, et riche aussi d'un autre enseignement quand on se trouve confronté à la complexité de l'articulation entre les aspects politiques, budgétaires et pédagogiques. Le suivi de l'expérience a donné lieu à plusieurs documents et rapports qui gardent trace de l'éveil des enfants, de leur réflexion et de leur progression. Si les instances de tutelle demandent bien souvent des rap-

ports d'évaluation écrits pour le suivi de telles expérimentations, le vidéogramme *Machin machine*²⁹ pourrait être considéré, à divers égards, comme un modèle de document essentiel pour de tels rapports. Caméra à l'épaule, filmant parmi les enfants, assis par terre ou à côté d'eux, l'intrus silencieux est rapidement oublié, et pourra ainsi montrer ce qu'il est si difficile d'imaginer quand on ne l'a pas vécu.

²⁹ Vaudène D. & Nowak G. (1984). « Machin machine ». Vidéo 26 mn. Production : Atelier de recherche du département d'audio-visuel de l'Université Paris VII (Expérimentations avec Logo au groupe scolaire Corbon-Alleray). Les photogrammes figurant dans ce texte sont extraits de cette vidéo.

Bibliographie

Berry G. (2019). *La pensée informatique*. Paris : CNRS Éditions.

Bouveresse J. (1980). « La philosophie et les fondements ». *Archives de Philosophie*, 43-1, pp. 3-32.

Chadwick I. (1984). « Logo's Lineage, The Canadian connection ». *Antic*, vol. 2, n° 12, mars 1984, pp. 28-31.

Clements D. H. & Meredith J. S. (1992). *Research on Logo: Effects and Efficacy* [en ligne]. New York, NY : Logo Foundation. [URL : https://el.media.mit.edu/logo-foundation/resources/papers/pdf/research_logo.pdf].

CNRS/LISH (dir.) (1984). « 2^e colloque Logo, Le Mans, 3-4 novembre 1983. Textes des contributions ». Dossier de *E.T.I. – Éducation télématique informatique*, n° 3.

Darche P. & Nowak G. (1992). « ActNet, a network of actors based on transputers for apprenticeship of parallelism ». *Proceedings of Transputers'92* (Arc-et-Senans, 20-22 mai 1992)³⁰. Amsterdam: IOS Press, pp. 293-306.

Derrida J. (1967). *De la grammatologie*. Paris : Minuit.

Nowak G. (1982). « Présentation de quelques "bugs" et remèdes dans la phase d'expérimentation de la pratique active de l'informatique en milieu scolaire primaire ». *Actes du premier colloque Logo* (Clermont-Ferrand, 9-11 décembre 1982), INRP et les IREM³¹ de Clermont-Ferrand et d'Orléans, pp. 45-53.

Nowak G. & Le Roch M.-F. (1984). « En amont de Logo. Apprentissages au cours moyen – Expressions » (Actes du séminaire « Logo et handicaps – L'informatique au service de l'éducation spécialisée », Centre national d'études et de formation pour l'enfance inadaptée). *Le Courrier de Suresnes*, n° 42, 1985, pp. 59-85.

Nowak G. & Le Roch M.-F. (1989). « Méta-Vlisp, langue maternelle pour acteurs robotiques ». *Actes du Premier congrès francophone de robotique pédagogique*. Le Mans : Laboratoire d'informatique Université du Maine, pp. 167-177.

Nowak G. & Le Roch M.-F. (1990). « Moteurs ! Acteurs ! ». *Actes du Deuxième congrès international de robotique pédagogique*. Montréal : Les publications de la Faculté des sciences de l'éducation, pp. 165-180.

Papert S., Bamberger J., Minsky M. & Solomon C. (1973). « Uses of technology to enhance education » [en ligne]. *AI Memo*, n° 298 MIT, AI Laboratory [URL : <https://dspace.mit.edu/handle/1721.1/6213>].

Papert S. (1980). *Mindstorms: children, computers and powerful ideas*. Trad. fr. Vassallo-Villaneau R.-M. (1981). *Jailissement de l'esprit : ordinateurs et apprentissage*. Paris : Flammarion.

Pissavin P. (2019). « De quoi les "théorèmes de limitation des formalismes" : Théorèmes de Gödel de 1931 et apparentés, sont-ils la limitation ? ». Thèse de doctorat en Histoire et Philosophie des sciences, soutenue à l'Université Paris I Panthéon-Sorbonne (HAL, tel-02493455).

Salanskis J.-M. (2011). *Le monde du computationnel*. Paris : Les Belles Lettres.

Simon J.-C. (1981). *L'éducation et l'informatisation de la société* (Rapport au Président de la République). Paris : La Documentation française.

Turing A. (1936). « On computable

³⁰ Le texte reprend l'essentiel d'un rapport interne d'expérimentation « En amont de Logo. Apprentissages au cours moyen – Communication, coordination » de Nowak G. & Le Roch M.-F. de 1985.

³¹ Institut de recherche pour l'enseignement des mathématiques.

numbers, with an application to the Entscheidungsproblem ». *Proceedings of the Mathematical Society*, série 2, vol. 42, pp. 230-265. Trad. fr. Girard J.-Y. (1995). In *La machine de Turing*. Paris : Seuil, pp. 47-104.

Vaudène D. (2016). « Condition de possibilité et effets de limitations dans les théories et les modèles » [en ligne]. *Eikasia*, n° 72 [URL : <https://revistadefilosofia.org/72-04.pdf>].

Vaudène D. (2017). « Dialectique des effets d'insu » [en ligne]. *Eikasia*, n° 78 [URL : <http://revistadefilosofia.org/78-03.pdf>].

Vaudène D. (2019). « Un acheminement vers la question de l'écriture ». *Intentio*, n° 1, pp. 217-289.

Vaudène D. (2021). « De l'information à l'écriture » [en ligne]. *Revue d'intelligibilité du numérique*, n° 2 [URL : <https://intelligibilite-numerique.numerev.com/numeros/n-2-2021/2624-de-l-information-a-l-ecriture>].

Vaudène D. & Nowak G. (1985). *Polylog – Mode d'emploi*. Vidéo, 40 mn. Production et diffusion : Agence de l'informatique et plan Informatique pour tous. (Présentation et installation d'un Nano-réseau à l'usage des enseignants d'écoles, de collèges et de Centres de formation d'apprentis).

Vivet M. (1979). « Un camp de vacances informatiques – l'expérience pédagogique d'Arc-et-Senans ». *L'ordinateur individuel*, n° 13, décembre 1979, pp. 29-36.

Walden D. & Nickerson R. (ed.) (2011). *A Culture of Innovation. Insider Accounts of Computing and Life at BBN*, 2nd edition [en ligne]. East Sandwich, MA : Waterside Publishing [URL : <http://walden-family.com/bbn/bbn-print2.pdf>].

Wing J. (2006). *Computational Thinking*. CACM vol. 49, n° 3, pp. 33-35.

Wittgenstein L. (1969). *Philosophische Grammatik*. Oxford : Blackwell.

Les auteurs de ce texte tiennent à remercier toutes les personnes – et particulièrement les élèves –, les institutions et les associations qui, de près ou de loin, ont contribué et participé à cette expérience, et l'ont rendue possible. À l'époque, les trois co-auteurs étaient en poste à l'Institut de programmation.

Archéologie d'un carton de l'Informatique pour tous

Jean-Noël Lafargue

Membre associé de l'équipe Théorie Expérimentation Arts Médias et Design (unité de recherche AIAC), Université Paris 8.

Membre de l'équipe Interactivité Design et Art (EsadHaR recherche), École supérieure d'art et design Le Havre/Rouen.

Les enseignants d'une école primaire du Havre, soucieux de faire un peu de place dans leurs placards, sont tombés sur un lot de matériel informatique qui n'avait pas été utilisé depuis près de trente ans. Ne sachant qu'en faire mais supposant qu'il pouvait avoir un intérêt historique, une professeure, par ailleurs mère d'un étudiant de l'école d'art du Havre où j'enseigne, a suggéré qu'il serve dans le cadre des travaux de mon atelier de design numérique.

Il s'agit de vestiges du plan Informatique pour tous (IPT), lancé sous le gouvernement Fabius en janvier 1985 et abandonné en 1989. Divers indices laissent penser que l'ensemble date de l'année même du lancement de l'opération, il y a trente-sept ans. Les actuels

vétérans de l'établissement ne sont là que depuis le début des années 2000 et n'ont jamais employé ce matériel, il n'y a donc là plus personne pour témoigner de l'application du plan Informatique pour tous.

Le lot tient dans une boîte en carton sans couvercle destinée à l'archivage, qui mesure 60 x 40 x 30 cm. Les divers éléments sont en état presque neuf, si ce n'est qu'ils sont pour la plupart marqués d'un coup de tampon ou d'une inscription au feutre indélébile qui indique le nom de l'école à laquelle il appartenait¹.

¹ Il s'agit en fait de deux écoles : l'école Jules Ferry et l'école Jean Zay, deux établissements situés à cent cinquante mètres l'un de l'autre qui ont fusionné en 2015. J'ignore si ces deux établissements ont toujours mutualisé leur équipement informatique.

Le groupe scolaire est situé à Sanvic, un quartier résidentiel de la partie haute de la ville du Havre. Mon enquête sur le contexte ne va pas au-delà de ces quelques éléments, ma découverte du contenu de la boîte restera donc assez descriptive.

L'élément central est un ordinateur personnel Thomson TO7/70. Produit entre 1984 et 1986 par la société Thomson Micro-informatique², remplaçant le TO7 (1982-1984) – le TO7/70 étant l'ordinateur qui a été le plus diffusé dans le cadre du plan Informatique pour tous. Il a été déployé (nous dit la brochure qui accompagne l'ensemble), dans 24 000 établissements scolaires. J'ignore si le groupe scolaire dont je tiens cet ordinateur a disposé de plusieurs TO7/70 employés en « nanoréseau »³.

² Thomson Micro-informatique avait été nationalisée en 1982 au sein du groupe Thomson, qui contenait aussi Thomson-Brandt (électroménager, devenue Fagor) et Thomson-CSF (électronique de pointe et communications sans fil, actuelle Thales). Le groupe Thomson a abandonné son activité informatique et Thomson Micro a déposé le bilan en 1989.

³ Le Nanoréseau est un concept central du plan IPT. Conçu à l'université de Lille et développé par la société Léanord, il permettait à un ordinateur « tête », de type IBM PC (typiquement d'une marque française telle que Léanord, Goupil ou Bull) de piloter 31 ordinateurs Thomson TO7 ou MOS.

Carton principal

Un ordinateur Thomson TO7/70, dans sa boîte d'origine

L'unité centrale, son lecteur de cartouches et son clavier forment un seul bloc. Une trappe permet de loger le stylo optique, dispositif de pointage relié à l'ordinateur par un cordon torsadé, qui constituait une des grandes particularités de cet ordinateur face à ses concurrents. À l'époque, quelques rares modèles d'ordinateurs (Atari ST, Commodore Amiga, Apple Macintosh) étaient équipés de souris, mais la grande majorité ne se pilotait qu'au clavier. L'ordinateur se raccorde à un moniteur ou à un simple téléviseur à l'aide d'une prise péritel. Cet ordinateur se vendait 3 500 francs, ce qui correspond à un peu moins de 1 000 euros actuels⁴. C'était plus cher que les très populaires Sinclair ZX81 (600 francs), Commodore 64 (1 800 francs) et Sinclair ZX Spectrum (2 000 francs), mais bien moins cher que l'Apple II (11 000 francs, moniteur compris). Les ordinateurs de l'époque avaient chacun leurs spécificités techniques, leur système d'exploitation dédié, et ils n'étaient compatibles entre eux que via des langages de programmation tels que le Basic et le Logo. Il est en bon état de marche, si ce n'est que son stylo optique, qui détermine ses

⁴ Les conversions des francs vers l'euro tiennent compte de l'inflation et se fient au convertisseur proposé par l'INSEE [URL : <https://www.insee.fr/fr/information/2417794>].

coordonnées sur l'écran en fonction de sa fréquence de balayage, ne peut fonctionner qu'avec un écran cathodique : il se révèle totalement inutile avec les actuels écrans à cristaux liquides.

Lecteurs de cassettes

Deux lecteurs de cassettes sont présents dans la boîte. Selon leurs emballages, l'un est destiné à être utilisé avec un ordinateur TO7 et l'autre avec un MO5, mais tous deux sont compatibles avec le TO7/70. Il ne s'agit pas de simples lecteurs de cassettes audio, car ils sont partiellement pilotables depuis l'ordinateur. Ces lecteurs, compatibles avec les « musicassettes » vendues dans le commerce pour l'enregistrement musical, servaient à stocker et à lire les programmes informatiques. C'était le mode de sauvegarde le plus répandu en micro-informatique à l'époque, devant les lecteurs de disquettes ou les disques durs.

Documentation logicielle

Tout d'abord, une *Documentation des logiciels Nathan écoles Français/Mathématiques* (grand format, 280 pages, éd. Cedic/Nathan 1985). En illustration sur la couverture, on voit un homme au chapeau haut-de-forme et à la veste rose qui enjambe des lettres et des chiffres. Ce type d'illustration évoque celles des ouvrages scolaires de l'époque mais l'intérieur est plus austère. Le contenu couvre tous les

niveaux d'école primaire, du cours préparatoire au cours moyen 2^e année. En introduction, un mode d'emploi détaille la procédure à exécuter pour charger les programmes dans l'ordinateur depuis un lecteur de cassettes. Les exercices de français sont des petits jeux : mots à deviner, à mémoriser, accords à compléter, vocabulaire à classer par famille. Les exercices mathématiques portent sur le calcul, la numération, le repérage dans le plan cartésien, etc., mais il on trouve aussi quelques jeux graphiques, notamment un programme de création de frises à l'aide de caractères et de symboles que l'on dispose avec différents types de symétries ou de répétitions. Des vignettes montrant les différents écrans sont reproduites dans le livre.

On trouve ensuite des documentations plus ou moins épaisses pour plusieurs logiciels. Les éditeurs sont Cedic/Nathan, To Tek international⁵, ACT Informatique/Hatier ou encore le Centre national de documentation pédagogique :

- *Colorcalc*, un tableur bureautique édité par Thomson, qui tire parti de l'interactivité permise par le crayon optique.
- *Colorpaint*, un logiciel de dessin qui autorise le dessin à main levée mais aussi le recours à des opérations diverses (symétrie, copie...). Notons que la définition d'affichage

⁵ To Tek était essentiellement un éditeur de jeux vidéo pour micro-ordinateurs.

- du TO7/70 était 320 x 200 pixels en 16 couleurs. L'argumentaire commercial est le suivant : « Avec Colorpoint, si vous avez un micro, vous savez dessiner ».
- *AGD/Gameco*, logiciel dont le but est apparemment de pouvoir définir le parcours d'un objet sur l'écran.
 - *Solfège/Techni-musique*, qui sert à l'apprentissage du solfège.
 - *Conjugaison*, un logiciel d'apprentissage de la conjugaison.
 - *Gérer vos fiches*, un logiciel rudimentaire de bases de données qui permet d'organiser ses recettes de cuisine ou sa collection de disques. L'exemple utilisé tout au long du manuel est un catalogue de bateaux de plaisance à moteur.
 - *Pique-Fiche*, un autre logiciel de gestion de base de données, spécifiquement dédié à l'éducation. Son intérêt majeur est de produire des représentations visuelles des données. Il s'agit en fait de l'adaptation en français du logiciel anglais *Picfile*. Le nom de son autrice signalé de manière ostentatoire dès la première page, il s'agit d'Anita Straker, pionnière de l'usage pédagogique des nouvelles technologies en Grande-Bretagne, qui a travaillé au sein du Department for Education dans les années 1970-1980.
 - *Matrix*, un programme de traitement matriciel de données qui fonctionne sur le principe du jeu *Mastermind*. Ses auteurs sont trois chercheurs français et la brochure contient en annexe les références de leurs publications scientifiques sur le sujet.
 - *Jacquard*, par Christian Andrieux, un outil d'« activité de programmation » qui, est-il expliqué, « fait partie de la génération des systèmes capables de comprendre les utilisateurs qui n'ont ainsi que peu d'efforts à fournir pour être compris de l'ordinateur, Jacquard étant là pour traduire la pensée des auteurs en des termes compréhensibles par les ordinateurs ». Les fonctionnalités sont nombreuses, tout comme les types de données manipulées (image, texte, musique). Apparemment le logiciel permet de créer des interfaces conversationnelles, mais le manuel est un peu confus, et malgré un bon quart d'heure à regarder tourner la cassette, je n'ai pas été en mesure de lancer le programme.
- Enfin, un classeur nommé « Documentation des logiciels école TO7/70 » qui contient des fiches pour chacun des logiciels énumérés plus haut, et pour la vingtaine d'autres inclus dans la « Valise logiciel » mais ne bénéficiant pas d'un manuel dédié ; ainsi qu'une feuille volante, signée du ministère, qui corrige les erreurs des manuels qui accompagnent la Valise logiciel TO7/70.

Manuels de programmation

Outre les logiciels, le carton contient plusieurs manuels de programmation, qui sont tous co-édités par To Tek et, sauf mention contraire, Cedic Nathan :

- *Basic sans peine / Auto-initiation au Basic MO5-TO7/70*. Agrémenté de dessins d'illustration par Henri Favre et Raoul Raba.
- *Basic, manuel de référence*. Ouvrage copieux qui détaille les commandes du Basic Microsoft. Le texte est en noir, les extraits de code en orange foncé (éd. To Tek).
- Un fascicule intitulé *Memento Basic*, en complément à l'ouvrage précédent (éd. To Tek / Microsoft).
- *Logo sans peine / Auto initiation au Logo MO5-TO7/70*, avec des illustrations clairement inspirées du dessinateur Moebius et signées « W. ».
- *Simulateur d'interpréteur Logo*, d'après les travaux du Centre Mondial informatique et ressource humaine⁶.

⁶ Le Centre Mondial de l'informatique, situé avenue de Matignon, voulu par François Mitterrand, fondé en 1981 par Jean-Jacques Servan-Schreiber et initialement dirigé par Nicholas Negroponte, fut un lieu assez unique qui entendait pallier la « fracture numérique » (on n'utilisait pas encore cette locution, mais l'idée était là) en participant à la diffusion du savoir informatique auprès des enfants, des jeunes, des demandeurs d'emploi, mais aussi du continent africain.

- *Initiation à Logo*, réalisé par l'équipe S.O.L.I. (Systèmes ordinateur Logo International). Seymour Papert, un des créateurs du Logo et son plus célèbre vulgarisateur, est remercié pour ses suggestions. Le dessin de couverture, signé Nadine Monnier, représente des tortues évoluant dans des nuages.
- *Logo – Manuel de référence*, réalisé par l'équipe S.O.L.I. (Systèmes ordinateur Logo International). Le dessin de couverture, signé Nadine Monnier, représente une tortue dans un labyrinthe.

Imprimés divers

À l'intérieur de la *Documentation des logiciels Nathan* est glissée une page pliée en deux, en bleu foncé sur papier glacé, qui contient le catalogue de l'éditeur, Cedic, une coentreprise des éditions Nathan et de la société Thomson, dédiée à la production d'ouvrages de référence sur l'informatique. Certains des ouvrages listés ont des sujets pointus et sans rapport avec des usages scolaires : Intelligence artificielle et systèmes experts, langages Pascal, Cobol, Lisp... Les tarifs font de 25 à 290 francs (7 à 81 euros).

Une brochure du même éditeur, datée de 1988, ne contient que des références liées aux apprentissages scolaires assistés par ordinateur : français, mathématiques, physique, biologie, histoire-géographie, langues.

Une seconde brochure, datée de 1987, est le catalogue de produits informatiques de la CAMIF (Coopérative des adhérents à la mutuelle des instituteurs de France). Le PC le moins cher, de marque Olivetti, est vendu 9700 francs (2500 euros), n'a pas de disque dur mais deux lecteurs de disquettes, et est fourni avec un écran. Le plus cher, un Léanord Elan AT Plus, fourni avec un disque dur de 20 Mo et une mémoire vive de 1 Mo est vendu 42670 francs (11000 euros).

Un manuel pour l'imprimante matricielle Mannesmann Tally MT80P, faisait probablement partie du kit fourni aux enseignants de l'école.

Enfin, le *Guide du maître TO7/70*, un fascicule d'une cinquantaine de pages édité par Cedic/Nathan et *a priori* fourni comme manuel (sans numéro d'ISBN), qui explique comment brancher l'ordinateur, comment y charger un programme, et qui expose sommairement la vocation des divers logiciels fournis.

Carton secondaire

Un deuxième carton contient vingt-huit cassettes, sur lequel est écrit en gros « Atelier "écoles" TO7/70 », et en plus petit : « Ministère de l'éducation nationale, mission aux technologies nouvelles, Atelier "informatique pour tous" ». Toujours sur le carton un cartouche signale que les cassettes doivent être stockées à température ambiante

(20 à 21 °C) pendant vingt-quatre heures avant d'être utilisées.

Chaque boîtier de cassette est blanc, avec pour seule indication un titre écrit en vert. Ces noms sont parfois explicites (*Solfège, Chiffres et lettres, Le corps humain, Grammaire...*), mais le plus souvent passablement obscurs : *Androïde, AGD, Memotex, Joligone, Chêne*.

Outre les vingt-huit cassettes de ce carton on trouve douze autres programmes, eux aussi sur cassette, présentés de la même manière, mais dont les titres sont écrits en rouge ou en bleu. Un boîtier en plastique au format d'une cassette VHS, nommé *Histoire de France*, contient deux cassettes correspondant chacune à une période de l'Histoire : -120 à 1450 et 1451 à 1980. Le contenu du programme est apparemment une frise chronologique.

Enfin, une autre boîte contient quatre cassettes et deux livrets d'une série nommée *En allant à l'école*, initiée par Cedic/Nathan et la Prévention routière, expliquent aux piétons (4-8 ans) et aux cyclistes (8-11 ans) comment se méfier des automobiles.

Le carton contient aussi des programmes sur cartouche⁷ : *Colorcalc* ; *Colorpaint* ; *Logo* ; *Générez vos fi-*

⁷ Les programmes sur cartouche se chargent de manière immédiate, contrairement aux cassettes audio. Ils sont en lecture seule (on ne peut pas les modifier).

chiers ; *Basic Microsoft 1/0* (en deux exemplaires). Sur la cartouche du Logo, un papier scotché signale, d'une écriture manuscrite : « *On ne voit pas la tortue, paraît-il, mais fonctionnait bien à la livraison* ».

La boîte contient en outre, en deux exemplaires, une brochure publiée par la Documentation pédagogique et le ministère de l'Éducation nationale.

Sur la couverture, des circuits imprimés lumineux en perspective forment un paysage blanc et ocre. Un dégradé qui passe du violet blanc puis au roux y fait office de ciel. Le titre, écrit en bleu clair avec la typographie Brush Script est *Informatique pour tous*. Plus bas, sont mentionnés les responsables du document : « Premier ministre, délégation chargée des nouvelles formations » et « Ministère de l'éducation nationale, mission aux technologies nouvelles ». Au dos, on voit l'image bucolique d'un paysage vallonné, avec village et clocher. Les photos intérieures, en noir et blanc, montrent : une classe d'école primaire, sans ordinateur visible mais avec un téléviseur, des enfants et une carte de France où apparaît en gros le nom « Massif central » ; une maison traditionnelle bretonne ; un homme âgé d'une trentaine d'années qui manipule un appareil électronique sous les yeux attentifs de collégiens ; le fronton d'une école communale de village ; un homme âgé qui manipule un clavier d'ordinateur sous les yeux attentifs d'un jeune homme ; quelques images de circuits intégrés.

L'ouvrage est préfacé par le premier ministre de l'époque, Laurent Fabius. Ce dernier commence par dire à quel point l'informatique prend une place importante dans tous les domaines de la vie quotidienne et professionnelle, avant de rappeler que cela ne garantit en rien « *une société de communication où tout le monde trouverait sa place de manière harmonieuse* ». Il avertit même que « *faute de s'y préparer, beaucoup de nos concitoyens risqueraient d'être marginalisés, ajoutant à une perte de bien-être individuel un manque à construire collectif* ». C'est cette ambition de ne laisser personne sur le bas-côté qui motive le plan Informatique pour tous, présenté le 25 janvier 1985. Le plan s'impose trois objectifs :

- Initier les élèves de toutes les régions de France à l'outil informatique.
- Permettre l'usage de l'outil informatique par tous les citoyens.
- Former 110000 enseignants en un an à l'utilisation des ateliers.

« *Point de départ d'une aventure nécessaire, il nous interpelle pour que cette génération soit la mieux formée de notre histoire* ».

Suivent quelques pages sur la vertigineuse accélération technologique et les prévisibles mutations de l'emploi et des formations que celle-ci imposera. La description de la mise en œuvre du plan passe en tout premier lieu par les « nano-

réseaux ». Les ordinateurs déployés sont essentiellement des Thomson TO7/70 (24 000 écoles) et des Exelvision EXL 100 (9 000 écoles). L'existence de modems est évoquée, mais pas vraiment leur usage. En 1985, les modems servaient surtout à connecter deux ordinateurs distants à l'aide d'une connexion téléphonique. Les écoles primaires, collèges, lycées et universités sont équipés d'ateliers en nanoréseaux ou parfois en « *nanoréseaux renforcés* » (ordinateurs plus puissants, logiciels professionnels, imprimantes). Les marques associées sont les fleurons de l'industrie informatique française : Thomson, Bull et Matra (nationalisées) mais aussi Logabax, Leanord et SMT Goupil (privées). La mise en œuvre n'est pas que matérielle. La formation des enseignants est bien entendu évoquée (un exemple d'emploi du temps d'une formation destinée aux enseignants est proposé), ainsi que la création ou le renforcement de diplômes liés aux professions de l'informatique (du CAP à la maîtrise). La possibilité de demander une aide à la création et à la diffusion de logiciels est aussi prévue.

Le reste de la brochure est constitué de ressources très concrètes, avec notamment une copie du décret qui a instauré le plan Informatique pour tous, un annuaire de centres de formation, de responsables académiques, de commissaires coordonnateurs du plan, de préfectures de région, de centres de documentation pédagogiques.

Le message d'ensemble est un rien menaçant : pour exister dans le monde à

venir, il sera obligatoire à chacun de se mettre au pli de l'informatisation, et ce ne sera ni amusant ni intéressant, juste nécessaire⁸. Mais heureusement, le péril est anticipé par le gouvernement qui déploie des moyens considérables pour s'assurer que chaque Français parvienne à passer ce cap. Les photographies en noir et blanc ramènent quant à elles à une France « éternelle », non-urbaine, et nous montrent systématiquement des aînés en position de transmettre un savoir aux plus jeunes – ce sont eux en tout cas qui tiennent les claviers.

Pliée en deux dans une des brochures, on trouve enfin une lettre (imprimée) signée par Jean-Pierre Chevènement, ministre de l'Éducation nationale en 1985, qui s'adresse aux directeurs et directrices d'écoles et leur dit sa certitude qu'avec l'ambitieux plan Informatique pour tous, « *notre pays se portera au premier rang des nations modernes* ».

⁸ On pense au très populaire ouvrage d'Alvin Toffler paru dix ans plus tôt, *Le Choc du futur* (Denoël, 1973 ; traduit de l'américain : *Future shock*, Bantam, 1970) qui formulait le même genre de menace : chacun doit s'adapter au progrès technologique, ou bien devenir obsolète.

Rôle d'un logiciel dans la transposition didactique du concept d'algorithme : le cas du logiciel *AlgoBox* en France et des programmes du lycée entre 2009 et 2019

Antoine Meyer

LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris

Simon Modeste

IMAG, Université de Montpellier, CNRS

Résumé

Cet article présente une analyse épistémologique et didactique de phénomènes curriculaires relatifs à l'enseignement de l'algorithmique en France pendant la période 2009-2019 dans le cursus de mathématiques du lycée. Notre étude s'appuie sur l'analyse du logiciel AlgoBox, proposé pour une initiation à l'algorithmique en cohérence avec ces programmes et très utilisé pendant la période. Ce logiciel présente des choix techniques marqués dont nous questionnons les conséquences. Nous montrons en quoi le « projet » incarné par AlgoBox se heurte à des difficultés d'ordre conceptuel et didactique.

Mots-clés : algorithmique ; programmation ; lycée ; *curriculum* ; didactique.

Abstract

This article presents an epistemological and didactical analysis of curricular phenomena regarding the teaching of algorithmics in France in 2009-2019 in the mathematics curriculum of high school. Our study is based on the analysis of AlgoBox, a software designed for an introduction to algorithmics in coherence with the curriculum, and much used during this period. This software makes marked technical choices, whose consequences are questioned here. We show how the “project” embodied by AlgoBox runs up against conceptual and didactical difficulties.

Keywords: algorithmics; programming; high school; *curriculum*; didactics.

Introduction

L'histoire de l'enseignement de l'informatique au lycée en France est marquée par différentes périodes, montrant des rapports divers à l'informatique, avec des postures liées aux évolutions techniques de l'époque (Baron & Bruillard, 2011). Ainsi, l'option informatique des lycées mise en place dans les années 1980 se présente essentiellement comme un enseignement de programmation. Les années 1990 sont marquées par l'arrivée sur le marché de « progiciels », dont la prise en main et l'usage pédagogique viennent remplacer l'enseignement de la programmation. C'est au début des années 2000 que le débat sur l'enseignement des concepts de la science informatique, de la programmation et de l'algorithmique reprend de l'ampleur et aux alentours des années 2010 que l'on peut noter le retour d'éléments d'algorithmique et de programmation au lycée général (à partir de 2009), ainsi que de premiers enseignements autonomes d'informatique (à partir de 2012).

Cet article se focalise sur la période 2009-2019, délimitée par deux événements particulièrement marquants dans l'histoire de l'enseignement de l'informatique au secondaire en France¹. Malgré quelques aménagements opérés au fil des années, cet enseignement nous semble

relever d'un point de vue particulier sur la discipline en général, et sur l'algorithmique en particulier.

L'année 2009 marque l'entrée au lycée d'éléments d'algorithmique dans les programmes de mathématiques des filières générales du lycée (en seconde en 2009, puis en première en 2010 et en terminale en 2011), à l'occasion d'une réforme initiée par Xavier Darcos et poursuivie par Luc Châtel, alors ministres de l'Éducation Nationale. L'introduction de ces concepts auprès d'enseignants disposant de connaissances très hétérogènes en informatique a donné lieu à l'élaboration de contenus tout à fait spécifiques. Les attentes de ces nouveaux programmes ont engendré des phénomènes de transposition didactique² (Chevallard, 1985) du concept d'algorithme et de l'activité algorithmique, qui ont déjà fait l'objet d'études en didactique (Modeste, 2012, 2020).

En 2019, autre jalon important, une réforme profonde du lycée général, incluant notamment la suppression des filières S, L et ES du lycée général, a entériné l'apparition de l'informatique au lycée en tant que discipline scolaire à part entière : la spécialité « Numérique et Sciences Informatiques » (NSI) proposée en première et en terminale. Pour autant, les éléments d'algorithmique et de programmation déjà présents dans les programmes de mathématiques n'en furent pas retirés. C'est cependant à partir de cette date que

¹ Travail réalisé avec le soutien financier de l'ANR (projet DEMaIn, référence <ANR-16-CE38-0006-01>). Cet article est accompagné de 3 annexes, accessibles en ligne sur le site de la revue qui apportent des précisions sur certains aspects de nos analyses [URL : <http://chc.hypotheses.org>].

² Nous présentons le concept de transposition didactique un peu plus loin.

le langage de programmation Python est officiellement prescrit comme langage de référence pour enseigner ces contenus, ce qui constitue une rupture par rapport aux années précédentes (où aucun langage spécifique n'était imposé).

La question du choix du premier langage de programmation à enseigner à des débutants est ancienne et complexe, et a donné lieu dans la littérature à d'innombrables publications, au point d'être qualifiée avec humour par Gal-Ezer et Harel de « *guerre culturelle* » (Gal-Ezer & Harel, 1998, p. 82, notre traduction). Un exemple d'étude récente portant sur un grand nombre d'élèves et comportant une revue de littérature sur le sujet peut être trouvé dans (Chen & al., 2019). Sans prétendre répondre à cette question, c'est à cet enjeu que nous allons nous intéresser, en étudiant l'influence sur les *curriculums* du logiciel *AlgoBox* et de son langage, conçus spécifiquement pour l'enseignement de l'algorithmique au lycée dans le contexte de la réforme de 2009.

Pour ce faire, nous avons étudié les contenus algorithmiques des *curriculums* de mathématiques de lycée de la période 2009-2019, des particularités du logiciel *AlgoBox*, et de la place qu'il a occupée dans l'écosystème curriculaire de la période.

Notre article s'organise en quatre parties :

- présentation du cadre conceptuel, de la méthodologie et du corpus ;

- présentation et analyse des *curriculums* et leurs évolutions entre 2009 et 2019 ;
- présentation et analyse du logiciel *AlgoBox* et de son langage, et ses particularités techniques ;
- analyse et discussion de la place et du rôle d'*AlgoBox* au sein du *curriculum*.

Afin d'analyser les contenus d'algorithmique (et plus généralement d'informatique) dans le *curriculum* prescrit de mathématiques au lycée, ainsi que les caractéristiques et effets du logiciel *AlgoBox*, nous prenons appui sur les concepts de transposition didactique et de transposition informatique. La présentation de ces outils théoriques nous amène à préciser des éléments d'épistémologie relatifs aux concepts étudiés, qui constitueront une référence pour l'analyse. Nous présentons ensuite la méthodologie de l'étude et le corpus mobilisé.

Cadrement théorique

Transposition didactique

Pour cette recherche, nous ferons principalement appel au concept de transposition didactique. Ce concept a été introduit en didactique des sciences par Chevallard (1985) et s'est fortement développé en didactique des mathématiques (Chevallard & Bosch, 2014), en particulier dans le cadre de la théorie anthropologique du didactique (Bosch & Gascón, 2014).

La transposition didactique est définie comme le phénomène de passage d'un savoir de référence (académique, professionnel...) dit savoir savant, au savoir à enseigner (transposition dite externe), puis du savoir à enseigner au savoir enseigné (transposition dite interne). Le savoir à enseigner est celui dont les institutions d'enseignement prescrivent l'apprentissage, de façon plus ou moins explicite (c'est le savoir défini par les programmes officiels dans le cas de l'enseignement obligatoire en France).

Le processus de transposition didactique est soumis à l'action de divers agents (scientifiques et sociétés savantes, inspecteurs, sociétés d'enseignants et syndicats, acteurs du monde professionnel, rédacteurs de manuels, producteurs de ressources...) qui constituent ce que la théorie anthropologique du didactique nomme la noosphère, et qui participe à sélectionner, découper, réorganiser, structurer, définir, apprêter le corpus du savoir à enseigner et le savoir enseigné.

En rendant compte de la distance qui sépare le savoir savant du savoir enseigné, le concept de transposition didactique permet d'exercer une vigilance épistémologique sur les phénomènes curriculaires :

En effet, alors que l'école vit sur la fiction consistant à voir dans les objets d'enseignement des copies simplifiées mais fidèles des objets de la science, l'analyse épistémologique, en nous permettant de comprendre ce qui gouverne l'évolution de la connaissance scientifique, nous aide à prendre conscience

de la distance qui sépare les économies des deux systèmes (Artigue, 1990, pp. 244-245).

Cette « économie » des deux systèmes a été théorisée dans une approche écologique du didactique (Artaud, 1998) qui permet d'explicitier les conditions et contraintes qui régissent la co-habitation des savoirs entre eux, et leur existence, apparition, disparition, évolution, dans les *curriculums*. Dans cet article, nous retiendrons pour nos interprétations l'idée d'un écosystème didactique dans lequel les savoirs cohabitent et évoluent selon des dynamiques curriculaires.

Transposition informatique

« Aux contraintes de la transposition didactique s'ajoutent, ou plutôt se combinent, celles de modélisation et d'implémentation informatiques : contraintes de la modélisation computable, contraintes logicielles et matérielles des supports informatiques de réalisation » (Balacheff, 1994, p. 364). Ainsi, l'informatisation des objets d'enseignement, au sein de logiciels n'est pas une simple traduction, mais produit la transformation de ces objets. Il en résulte des objets d'enseignement possédant certaines représentations, propriétés et permettant ou limitant certaines opérations.

Ce phénomène de transposition informatique nous semble pertinent pour notre problématique au sens où les logiciels et langages de programmation utilisés

ont une implication directe sur la perception des concepts disciplinaires sous-jacents (ici algorithme et programme).

Repères épistémologiques

Dans une préoccupation de vigilance épistémologique mentionnée précédemment, nous présentons un ensemble de points d'attention sur lesquels s'articuleront nos analyses des *curriculum*s et du logiciel *AlgoBox*.

Nous nous appuyons sur des définitions et concepts issus directement du savoir savant, ou d'analyses épistémologiques antérieures, en particulier en ce qui concerne les concepts de problème algorithmique, d'algorithme et de programme, sur celui de spécification, de statut fonctionnel des variables, et enfin sur des considérations relatives à la représentation et à la sémantique des programmes.

Les éléments issus de ces considérations épistémologiques et de l'analyse du savoir savant peuvent être vus comme l'explicitation d'un modèle épistémologique de référence au sens de la théorie anthropologique du didactique (Bosch & Gascón, 2014), et serviront de référence pour nos analyses.

- *Algorithme et problème algorithmique*

Dans sa thèse, en appui sur plusieurs ouvrages de référence, Modeste (2012) propose une définition générale du concept

d'algorithme reposant sur le concept important de « problème algorithmique », qu'il convient de distinguer du concept général de problème rencontré en mathématiques, en science ou en didactique.

Un problème p est un couple (I, Q) , où I est l'ensemble des instances (pouvant être décrit par plusieurs paramètres) du problème et Q une question portant sur ces instances (spécifiant les propriétés de la solution attendue). [...]

Un algorithme est une procédure de résolution de problème, s'appliquant à une famille d'instances du problème et produisant, en un nombre fini d'étapes constructives, effectives, non-ambigües et organisées, la réponse au problème pour toute instance de cette famille (Modeste, 2012, p. 25).

L'expression « résolution de problème » est à entendre ici dans un sens plus concret et opératoire que dans le sens général de « démarche de résolution de problèmes » utilisé parfois dans le milieu de l'éducation pour faire référence à un certain type de compétences d'investigation, de raisonnement ou d'expérimentation attendu des élèves. Nous reviendrons sur cette polysémie du mot « problème » plus loin.

Nous insistons également sur les notions d'« instance » d'un problème (potentiellement définie par plusieurs paramètres), et sur celle de « résultat », composantes essentielles de la notion stricte d'algorithme. Dans une pratique experte, il est attendu que la présentation d'un algorithme s'accompagne de sa « spéci-

fication », qui consiste généralement en la description *a priori* du problème que l’algorithme prétend résoudre. Cette exigence s’étend également à la pratique de la programmation.

- *Programmes et machines*

Comme nous l’avons vu, un algorithme est un objet de nature abstraite dont la définition ne fait pas nécessairement référence à un quelconque dispositif physique de calcul³. Il est cependant entendu que la raison d’être d’un certain nombre d’algorithmes est d’être traduits (ou « implémentés ») en un « texte » appelé programme destiné à être exécuté par une telle « machine », par exemple par un ordinateur.

Nous définissons un programme comme une suite de symboles (un texte), agencée selon une grammaire ou syntaxe prédéfinie, dont l’interprétation obéit à une certaine sémantique tenant compte des capacités et contraintes de la machine (réelle ou virtuelle) à laquelle il est destiné. Dans Durand-Guerrier & al. (2019), où nous explorons l’articulation entre syntaxe et sémantique dans une perspective de didactique des mathématiques et de l’informatique, nous précisons cette distinction dans les termes suivants :

Dans ce contexte [en informatique],
« syntaxe » fait référence aux règles

³ On peut néanmoins faire appel à une notion de « machine abstraite » afin de caractériser formellement le sens d’un algorithme, notamment pour pouvoir étudier sa complexité. Nous ne mobiliserons pas ce concept dans cet article.

de composition d’un texte valide dans le langage de programmation choisi, et « sémantique » à l’effet attendu de chaque élément syntaxique de ce langage et des combinaisons de ces éléments sur la machine réelle (ou sur un modèle de la machine) (Durand-Guerrier & al., 2019, p. 122, notre traduction).

On nommera dans ce texte « langage de programmation » un système cohérent formé d’un ensemble de symboles, d’une syntaxe et d’une sémantique (en référence à une certaine machine). Nous ne détaillerons pas plus ces définitions, qui font l’objet de discussions plus approfondies dans le chapitre cité.

Il nous paraît également important d’insister sur la différence entre algorithme et programme, comme le résume clairement cet extrait de (Froidevaux, Gaudel & Soria, 1990, p. 6) :

Un algorithme doit être exprimé dans un langage de programmation pour être compris et exécuté par un ordinateur. Cependant, il faut bien comprendre qu’un algorithme est indépendant du langage de programmation utilisé. L’algorithme d’Euclide programmé en Pascal, en Ada, ou en Lisp, reste toujours l’algorithme d’Euclide.

- *Représentation des algorithmes et programmes*

Dans une perspective d’enseignement de l’algorithmique et/ou de la programmation, particulièrement à desti-

nation de débutants, le choix d'un mode de représentation des algorithmes (textuelle ou graphique, plus ou moins formalisée) est une question importante⁴.

Dans une étude épistémologique à visée didactique, Nguyen (2005) étudie les choix de langage ou de machine de référence dans des traités d'initiation à l'algorithmique et à la programmation, et identifie deux stratégies principales (vues comme des façons d'organiser le corps des savoirs dans un enseignement) :

Première stratégie : présence d'une machine de référence, réelle ou fictive

Ces machines fictives peuvent être virtuelles [...], ou idéales [...]. L'enseignement de l'algorithmique et de la programmation est alors lié à la compréhension de l'architecture de la machine et à la conception d'un langage orienté vers cette machine.

Deuxième stratégie : absence de machine de référence

Dans cette stratégie, l'enseignement de l'algorithmique passe par celui d'un langage de programmation représentatif d'une classe de langages existants [...]. Cet enseignement suppose la connaissance préalable d'un langage de programmation (Nguyen, 2005, p. 58).

Pour caractériser les représentations des algorithmes, Modeste (2012, 2020) propose quant à lui trois para-

digmes que sont la preuve algorithmique (les algorithmes restent implicites dans les constructions mathématiques et les preuves), l'algorithmique mathématique (représentation des algorithmes dans un langage spécifique mélange de langage mathématique et de mots-clés liés aux structures de contrôle ou inspirés de langages de programmation, parfois appelé pseudo-langage) et algorithmique informatique (représentation des algorithmes dans des langages informatiques, formels, ayant une syntaxe et une sémantique bien définies).

Ce que nous retenons des deux points de vue de Nguyen et Modeste est l'importance de prendre en compte les représentations des algorithmes pour comprendre les positionnements épistémologiques et la transposition didactique en jeu. La question des effets sur l'enseignement et l'apprentissage de l'algorithmique du choix d'un langage de programmation ou d'une représentation semi-formelle d'algorithmes telle que le « pseudo-code », constitue un thème de recherche à part entière qui dépasse le périmètre de ce travail.

• *Statuts fonctionnels des variables*

Le concept de variable est central dans l'apprentissage de la programmation et, dans une certaine mesure, de l'algorithmique. Samurçay (1985) étudie les conceptions et difficultés d'élèves au sujet des variables pendant la phase d'alphabetisation, comme elle appelle la période de première initiation à l'informatique. Elle

⁴ Voir par exemple (Chen & al., 2019) au sujet des impacts différents des langages textuels et par blocs sur l'apprentissage.

identifie en particulier deux statuts fonctionnels de la variable :

Dans les situations-problèmes que rencontre effectivement le sujet, les variables interviennent avec des statuts fonctionnels différents. Dans un premier temps on peut en distinguer deux :

- les variables qui sont les données explicites du problème ;
- les variables qui sont rendues nécessaires par la solution informatique (Samurçay, 1985, p. 146).

Nous reprenons à notre compte cette catégorisation, en particulier dans un contexte de programmation impérative, et la complétons par un troisième statut fonctionnel correspondant aux variables destinées à désigner le résultat final d'un algorithme (l'usage d'une ou plusieurs variables ayant ce statut n'est pas systématique mais apparaît dans l'écriture de nombreux algorithmes). Par commodité, nous désignerons dorénavant ces trois statuts sous les noms de « variables-paramètres », « variables de travail » et « variables-résultat ».

• *Résolution de problèmes et pensée informatique*

Wing (2006) définit la pensée informatique comme une manière possible pour les humains (et non les ordinateurs) de résoudre des problèmes, dans un sens du mot « problème » plus général que le strict sens algorithmique évoqué ci-dessus. Elle caractérise la pensée algorithmique comme constituée

de plusieurs méthodes et formes de raisonnement, qui ont ensuite fait l'objet de débats. Dans une revue de littérature à ce sujet, Selby et Woollard (2013) tentent de repérer et nommer les caractéristiques de la pensée informatique qui font consensus à la fois du point de vue du savoir savant et au sein de la recherche sur l'enseignement de l'informatique. Ils en dégagent ainsi plusieurs composantes propres, invariantes, consensuelles et bien définies, que sont : la capacité à raisonner en termes de décompositions, à l'aide d'abstractions, de manière algorithmique, en termes d'évaluations et en termes de généralisations. Nous renvoyons le lecteur au texte de Selby et Woollard pour une définition plus détaillée de ces composantes.

Dans ce travail, nommer et distinguer différentes composantes de la pensée informatique nous permettra de poser la question des types de raisonnements que les transpositions didactique et informatique à l'œuvre dans les programmes et dans *AlgoBox* permettent ou entravent.

Méthodologie et corpus

Pour approfondir les questions présentées dans la problématique, nous allons conjointement étudier les *curriculums* d'algorithmique de la période 2009-2019 et présenter une analyse technique et didactique du logiciel *AlgoBox*, l'objectif étant de montrer comment ont co-évolué le logiciel et les *curriculums*.

Dans un premier temps nous porterons un regard didactique sur l'histoire récente de l'enseignement de l'algorithmique dans les programmes de mathématiques, et de l'enseignement de l'informatique dans l'enseignement secondaire en France. Ceci permettra de contextualiser la situation de la période 2009-2019, de détailler les principaux événements institutionnels liés aux *curriculums*, et d'identifier les principales caractéristiques de la transposition didactique externe du concept d'algorithme à l'œuvre.

Dans un second temps, nous étudierons en détail le logiciel *AlgoBox*, en menant une analyse des choix techniques et didactiques qui l'ont porté, ainsi qu'une identification et analyse de ses versions et de leur chronologie. Ceci nous permettra de caractériser la transposition informatique du concept d'algorithme réalisée par le logiciel.

Enfin, nous tenterons de montrer la place et le rôle qu'*AlgoBox* a pu jouer dans le *curriculum* « réel » au cours de la période étudiée, en d'autres termes son impact possible sur la transposition didactique interne des concepts en jeu, en mettant en perspective les spécificités du logiciel et celles des programmes de mathématiques. Ces analyses seront menées selon les repères épistémologiques ci-dessus.

Pour développer cette étude nous sommes appuyés sur les documents institutionnels témoignant du savoir à enseigner (programmes, documents-res-

sources), et du savoir enseigné (manuels scolaires, sujets de baccalauréat), des documents attestant des échanges et débat au sein de la noosphère (rapports d'experts, consultation, réactions...), ainsi que sur le logiciel *AlgoBox* lui-même (site officiel, documentation, code source, historique des versions, recueils d'activités).

Regard didactique sur l'histoire récente de l'enseignement de l'informatique dans les programmes de mathématiques du lycée

Panorama de l'enseignement de l'algorithmique au lycée (2009-2019)

- *Éléments précurseurs*

La Commission de Réflexion sur l'Enseignement des Mathématiques (CREM), créée en 1999, confiée à la direction du mathématicien et académicien Jean-Pierre Kahane⁵ et composée de chercheurs en mathématiques, informatique, didactique des mathématiques, et d'enseignants, formateurs et inspecteurs, produisit divers rapports entre 1999 et 2002 dont une compilation, le rapport Kahane, fut réalisée en 2002 (Kahane, 2002). Parmi les recommandations issues des premières

⁵ Duperret J.-C., « Histoire de la C.R.E.M. (1996-1999) », site *Educmath*, 2007 [URL : <http://educmath.ens-lyon.fr/Educmath/ressources/etudes/crem/>].

années de la commission figure le rapport « Informatique et enseignement des mathématiques » dans lequel est proposée l'introduction d'une part d'informatique dans l'enseignement des sciences mathématiques et dans la formation des maîtres.

Dans la lignée de ce rapport, les programmes de lycée mis en place entre 2004 et 2005 ont introduit, dans les classes de première et terminale littéraire uniquement, un domaine « algorithmique » (en tant que domaine transversal, tout comme le domaine « logique »). On peut considérer cette introduction comme une expérimentation, l'enseignement mathématique dans les classes littéraires étant soumis à moins de contraintes. La réforme initiée en 2009 a repris un certain nombre d'éléments de ces programmes, en étendant cette introduction à toutes les filières du lycée général et à la seconde.

- *Réforme des programmes du lycée de 2009-2012*

En mars 2009, en vue d'une réforme des programmes du lycée général, est mise en place une consultation pour des programmes de mathématiques de seconde contenant une part d'algorithmique (*op. cit.*, note 15). En juin 2009 paraît un document-ressource en algorithmique pour la classe de seconde et fin juillet 2009 paraissent les programmes définitifs de seconde, à appliquer à la rentrée de septembre 2009⁶. Nous

⁶ Les programmes du Ministère de l'Éducation nationale (MEN) datant de 2009 sont disponibles sur le site Eduscol.education.fr : « Programme d'enseignement

n'analyserons pas en détail les réactions et évolutions qui ont eu lieu durant la période, très courte, de consultation⁷.

L'introduction de l'algorithmique a été un point de débat important dans les communautés liées à l'enseignement des mathématiques (associations d'enseignants, sociétés savantes, Commission Française pour l'Enseignement des Mathématiques, réseaux des Instituts de Recherche sur l'Enseignement des Mathématiques...).

Dans le projet en consultation, la partie algorithmique constituait une section à part entière du programme et se basait fortement sur le calcul sur les nombres entiers⁸, cette entrée « Calcul sur les nombres entiers » a été supprimée à l'issue de la consultation, et l'algorithmique a retrouvé un statut transversal similaire à celui des programmes de

de mathématiques de la classe de seconde générale et technologique » (*BO n° 30 du 23/07/2009*) [URL : <https://www.education.gouv.fr/bo/2009/30/mene0913405a.htm>]. Voir aussi les documents-ressources, suggestions d'activité à destination des enseignants : « Ressources pour la classe de seconde – Algorithmique » [URL : https://cache.media.eduscol.education.fr/file/Programmes/17/8/Doc_ress_algo_v25_109178.pdf].

⁷ On peut trouver une trace de cette consultation ainsi que des réactions qu'elle a pu provoquer sur le site du café pédagogique [URL : http://www.cafepedagogique.net/lemensuel/lenseignant/sciences/maths/Pages/2009/102_ALAUNE.aspx] ; [URL : <http://www.cafepedagogique.net/lexpresso/Pages/2009/05/Nxprogrammesmathsconsultation.aspx>].

⁸ « Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul, en opérant essentiellement sur des nombres entiers » (p. 9).

2004-2005, ce qui constitue une modification importante du projet initial.

Notons également que les parties d'algorithmique des programmes de première et de terminale, parus respectivement en 2010 et 2011, sont essentiellement identiques à celles des programmes de seconde de 2009. Nous n'y reviendrons pas et ferons référence à « la réforme de 2009 » pour englober l'ensemble de ces programmes⁹.

Les attendus en termes d'apprentissage sont relativement modestes. Les élèves sont entraînés à « *décrire certains algorithmes en langage naturel ou dans un langage symbolique* », à en réaliser certains « *à l'aide d'un tableur ou d'un petit programme* » et à « *interpréter des algorithmes plus complexes* ». Les capacités attendues en fin de cycle concernent, par exemple, la capacité à écrire des « *instructions élémentaires (affectation, calcul, entrée, sortie)* » et à programmer des structures de « *boucle et itérateur, instruction conditionnelle* », le tout « *dans le cadre d'une résolution de problèmes* ».

Une analyse détaillée de ces programmes, dont nous proposons une relecture synthétique dans la section suivante, est disponible dans (Modeste, 2012). Elle fait apparaître en particulier un certain

amalgame entre les concepts d'algorithme et de programme, sur lequel nous reviendrons à plusieurs reprises dans cet article.

- *Spécialité « Informatique et sciences du numérique » (ISN)*

À la rentrée 2012, en même temps que se met en place le nouveau programme de terminale, est créée un enseignement de spécialité de Terminale Scientifique intitulé Informatique et Sciences du Numérique¹⁰, que l'on peut considérer comme une introduction de l'informatique en tant que discipline scolaire au lycée. Cette introduction a lieu dans un contexte où les communautés de recherche et professionnelle en informatique militent pour la mise en place d'un tel enseignement. En 2013, paraît un rapport de l'Académie des Sciences (2013) appelant à une généralisation de l'enseignement de l'informatique.

- *Réforme du collège de 2016*

Une réforme des programmes de collège est entrée en vigueur à la rentrée 2016. L'informatique comme discipline scolaire y apparaît explicitement pour la première fois, son enseignement, au collège, est dispensé à la fois en mathématiques et en technologie. Au cycle 4, les contenus relevant du domaine informatique apparaissent essentiellement

⁹ De même, pour une partie des citations des programmes, lorsque nous ne précisons pas de page, c'est que les extraits sont identiques dans tous les programmes de la période, de la seconde à la terminale.

¹⁰ Programmes MEN : « Enseignement de spécialité d'informatique et sciences du numérique de la série scientifique – classe terminale » (BO spécial n° 8 du 13 octobre 2011) [URL : <https://www.education.gouv.fr/bo/11/Special8/MENE1119484A.htm>].

dans les thèmes « L'informatique et la programmation » en Technologie, et « Algorithmique et programmation » en Mathématiques¹¹.

En mathématiques, ce nouveau programme met explicitement l'accent sur l'activité de programmation proprement dite, vu comme un thème à part entière du programme, non nécessairement lié aux autres thèmes mathématiques, et en distinguant les termes algorithme et programme. Comme cela apparaît clairement dans les recommandations officielles (voir en particulier celles de 2016¹²), il est proposé d'initier en priorité les élèves à la programmation à l'aide d'un langage de programmation événementiel par blocs, au premier rang desquels le logiciel *Scratch*¹³ fait figure de langage de référence, ce qui constitue une autre différence importante avec les programmes du lycée.

- *Aménagement des programmes du lycée de 2017*

En mai 2017, suite à la mise en œuvre simultanée de la réforme sur les

trois années du cycle 4, est paru un aménagement des programmes de mathématiques de seconde, pour une mise en œuvre dès la rentrée 2017 et une stabilisation prévue en 2019¹⁴. Tout comme en cycle 4 et en contraste avec les programmes précédents du lycée, il y est explicitement fait mention de programmation, et le domaine Algorithmique et programmation gagne le statut de thème à part entière. Le document-ressource accompagnant cet aménagement des programmes précise cependant : « À la différence du programme de mathématiques du cycle 4 du collège, il s'agit donc d'adosser explicitement les activités de la partie algorithmique et programmation aux mathématiques »¹⁵, ce que confirment les exemples traités, allant bien au-delà du niveau de la classe de seconde, orientés essentiellement vers l'analyse numérique et la simulation en probabilités et statistiques.

Le programme s'appuie sur « deux idées essentielles : la notion de fonction d'une part, et la programmation comme production d'un texte dans un langage

11 Documents-ressources MEN 2015 : « Programme d'enseignement du cycle des approfondissements (cycle 4) » (*BO* spécial n° 11 du 26 novembre 2015) [URL : <https://www.education.gouv.fr/bo/15/Special111/MENE1526483Aannexe3.html>].

12 Documents-ressources MEN 2016 : « Algorithmique et programmation » [URL : https://cache.media.eduscol.education.fr/file/Algorithmique_et_programmation/67/9/RA16_C4_MATH_algorithmique_et_programmation_N.D_551679.pdf].

13 Site du logiciel *Scratch* [URL : <https://scratch.mit.edu/>].

14 Documents-ressources MEN 2017 : « Ressources pour le lycée – Mathématiques – Algorithmique et programmation » [URL : https://cache.media.eduscol.education.fr/file/Mathematiques/73/3/Algorithmique_et_programmation_787733.pdf]. Par contraste avec les langages de programmation par blocs utilisés au cycle 4. Voir le programme MEN 2017 : « Seconde générale et technologique – Aménagements des programmes d'enseignement de mathématiques et de physique-chimie », p. 10 (*BO* n° 18 du 4 mai 2017) [URL : <https://www.education.gouv.fr/bo/17/Hebdo18/MENE1712512C.htm>].

15 *Ibid.*, p. 1.

informatique d'autre part »¹⁶. À l'opposé de l'approche non prescriptive des programmes de 2009-2012, il introduit également des consignes précises concernant le choix d'un langage de programmation :

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes. Le choix du langage se fera parmi les langages interprétés, concis, largement répandus, et pouvant fonctionner dans une diversité d'environnements (*ibid.*, p. 10).

Cette description désigne sans le nommer le langage Python, déjà très répandu dans les communautés éducatives de nombreux pays (y compris dans l'enseignement supérieur français) et utilisé de manière explicite (et exclusive) dans le document-ressource¹⁷.

- *Aménagements transitoires pour le bac 2018*

En 2017, en réponse à la situation transitoire que produit l'aménagement des programmes, un document diffusé par les canaux officiels propose une modification de l'écriture des algorithmes au baccalauréat¹⁸. Ce document sans en-tête

ni présentation, propose la réécriture de 5 algorithmes donnés au bac 2017 selon de nouvelles conventions, explicitant pour chaque cas les nouvelles attentes. On peut y lire une volonté de structurer le langage de description des algorithmes. Parmi les changements notables on note une « simplification de la syntaxe » (expression utilisée à de nombreuses reprises) qui se traduit par :

- La suppression de la déclaration de variable, remplacée par une précision dans l'énoncé des « hypothèses faites sur les variables » et le fait de « renoncer aux entrées sorties » (les instructions « Saisir » et « Afficher » disparaissent). Ainsi, par exemple, S étant une variable saisie et λ une variable affichée dans le bloc « Sortie », on ne pose plus la question « Quelle valeur affiche cet algorithme si on saisit la valeur $S = 0,8$? » mais « Si la variable S contient la valeur 0,8 avant l'exécution de cet algorithme, que contient la variable λ à la fin de son exécution ? » ;
- La suppression des rubriques « entrée », « variables », « initialisation », « traitement » et « sortie » ;
- Le remplacement des instructions « prend la valeur » ou « affecter à » par le symbole « \leftarrow » ;
- Le remplacement de mots en français par des symboles mathématiques (par exemple « supérieur à 1 » devient « ≥ 1 »).

¹⁶ Programme MEN 2017 : « Seconde générale et technologique », *op. cit.* (note 14).

¹⁷ Documents-ressources MEN 2017 : « Ressources pour le lycée », *op. cit.* (note 14).

¹⁸ « Évolution de l'écriture des algorithmes au baccalauréat ». Ce document n'a pas de statut très clair, mais on en retrouve de nombreuses instances sur des sites académiques tels que le site disciplinaire de mathématiques de l'académie de Créteil – mis en ligne le 18/10/2017 [URL : <https://maths.ac-creteil.fr/spip.php?article232>].

On peut noter que les mots-clés « Fin Pour » et « Fin Tant que » sont conservés, et que, malgré tout, il n’y a pas complète uniformité de langage dans les algorithmes réécrits.

- *Réforme du lycée de 2019–2020*

Suite à une consultation initiée à la fin de l’année 2017 et pendant une partie de 2018, de nouveaux programmes de seconde et première du lycée général et technologique ont été publiés en janvier 2019, puis pour la terminale en juillet 2019¹⁹. Cette réforme a amené à la disparition des séries S, ES et L du lycée général, au profit d’un choix de matières dites « de spécialité » par les élèves : trois matières d’un volume de quatre heures par semaine en première, et deux d’un volume de six heures en terminale.

Le thème « Algorithmique et programmation » de seconde conserve essentiellement le même contenu que celui de 2017. Il précise les contenus visés (variables, instructions et structures de contrôle élémentaires, fonctions) et prescrit Python comme langage de référence. Le

¹⁹ Programmes MEN 2019 : « Programme d’enseignement de mathématiques de la classe de seconde générale et technologique » (BO spécial n° 1 du 22 janvier 2019) [URL : <https://www.education.gouv.fr/bo/19/Special11/MENE1901631A.htm>] ; « Programme d’enseignement de spécialité de mathématiques de la classe de première de la voie générale » (BO spécial n° 1 du 22 janvier 2019) [URL : <https://www.education.gouv.fr/bo/19/Special11/MENE1901632A.htm>] ; « Programme d’enseignement de spécialité de mathématiques de la classe terminale de la voie générale » (BO spécial n° 8 du 25 juillet 2019) [URL : <https://www.education.gouv.fr/bo/19/Special18/MENE1921246A.htm>].

programme mentionne explicitement les « types de variables » à savoir utiliser : booléens, entiers, flottants et chaînes de caractères. Les listes sont introduites en première générale, pour les élèves ayant choisi la spécialité Mathématiques. Aucune notion nouvelle n’est introduite en terminale.

Deux matières d’informatique proprement dite ont été créées simultanément, un enseignement obligatoire « Sciences numériques et technologie » (SNT) en seconde, et une spécialité « Numérique et sciences informatiques » (NSI) en première et terminale, élargissant considérablement la portée de l’ISN de 2012. L’analyse des programmes de ces deux enseignements, dont l’un relève plutôt de la culture numérique et l’autre de science informatique, ne sera pas détaillée ici.

Analyse de la transposition didactique

Cette section propose une analyse de la transposition didactique à l’œuvre dans le *curriculum* de mathématiques de lycée au cours de la période étudiée, en appui sur les travaux précédents du domaine, notamment (Modeste, 2012, 2020), et en appui sur les éléments épistémologiques de la deuxième partie.

- *Présentation et définition du concept d’algorithme*

Le programme de seconde de 2009 ne définit pas formellement le concept d’algorithme, mais en donne quelques exemples

supposés avoir été rencontrés par les élèves au collège (« algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie »), ainsi que quelques aspects généraux : « Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul »²⁰.

Dans ce passage, les concepts d'entrées-sorties et d'affectation sont identifiés comme constitutifs, et même caractéristiques, de celui d'algorithme. Ces concepts, qui relèvent plutôt d'une certaine pratique de la programmation (en particulier de style dit *impératif*), ne sont pas forcément essentiels à l'activité ou au raisonnement algorithmique eux-mêmes. Il est difficile de déterminer quelles références sont à l'origine de cette conception de l'algorithmique, même si l'on en trouve des traces dans d'autres écrits francophones des années 1980 (voir par exemple Samurçay, 1985).

Le document-ressource lié²¹ donne une description légèrement plus détaillée du concept d'algorithme tel qu'il est envisagé en classe de seconde. Il s'appuie en particulier sur une définition attribuée à l'*Encyclopedia Universalis* :

On définit parfois les algorithmes de la manière suivante : « un algorithme

est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat et cela indépendamment des données. » Le résultat doit donc s'obtenir en un temps fini²².

Cette définition initiale est ensuite détaillée dans le document, en portant une attention particulière aux « algorithme[s] simple[s] », c'est-à-dire « ne comportant pas de fonctions ou sous-programmes »²³. Selon les auteurs du document, la conception d'un tel algorithme obéit à trois grandes étapes : 1) préparation du traitement (incluant le repérage, l'agencement et éventuellement l'entrée des données, et même le repérage de données intermédiaires), 2) traitement proprement dit, et 3) sortie des résultats.

Dans le programme de mathématiques de cycle 4 mis en application à partir de la rentrée 2016, un accent est mis sur certains aspects alors absents des programmes de lycée. Par exemple, les tout premiers attendus de fin de cycle mentionnés sur le thème Algorithmique et programmation sont « décomposer un problème en sous-problèmes afin de structurer un programme », et « reconnaître des schémas ». Comme évoqué à la section précédente, ces deux compétences sont identifiées par la littérature comme caractéristiques de la pensée informatique, et sont essentiellement absentes des textes encadrant les pro-

²⁰ *Op. cit.*, p. 9 (note 6).

²¹ « Ressources pour la classe de seconde – Algorithmique », *op. cit.* (note 6).

²² *Ibid.*, p. 6.

²³ *Ibidem.*

grammes de lycée, de par la conception de l'algorithme qu'ils véhiculent.

Suite à l'aménagement des programmes de 2017, on constate une très nette évolution dans la façon dont le concept d'algorithme est abordé. On trouve par exemple dans le document-res- source²⁴ la définition suivante :

Un algorithme est une procédure de résolution de problème, abstraction faite des caractéristiques spécifiques qu'il peut revêtir. Par exemple, un algorithme de tri ne résout pas le problème du tri d'un jeu particulier de données mais a pour objectif de trier n'importe quel jeu de données : le problème du tri s'applique à différentes instances, c'est-à-dire à différents jeux de données.

Un algorithme s'applique donc à une famille d'instances d'un problème et produit, en un nombre fini d'étapes constructives, effectives, non-ambigües et organisées, la réponse au problème pour toute instance de cette famille²⁵.

Il est intéressant de comparer cette définition à celle proposée par Modeste (2012) et citée dans la section précédente. Il est significatif que ce document officiel reprenne à son compte l'idée fondamentale de problème algorithmique, et fasse apparaître très clairement la notion d'instance d'un problème et celle de résultat. L'objet algorithme prend ici un sens mathématique précis, contrairement à la définition informelle qui prévalait depuis 2009.

• Algorithmique et programmation

Dans les *curriculums* mis en place en 2009-2012, au sujet de l'« entrée des données », il est précisé qu'elle « peut se manifester par la saisie de caractères ou de nombres sur le clavier, ou la lecture de la position du pointeur de la souris, ou encore par la lecture d'un fichier contenant ces nombres ou caractères »²⁶. La description de l'étape de « sortie des résultats » nous paraît également significative :

Les résultats obtenus peuvent être affichés sur l'écran, ou imprimés sur papier, ou bien encore conservés dans un fichier. Si on n'en fait rien, ils « restent » en mémoire jusqu'à la prochaine exécution ou sont perdus. À l'occasion, la sortie pourrait être graphique (afficher ou déplacer le pointeur de la souris ou des objets sur l'écran) ou sonore... voire sur Internet²⁷.

On constate que les auteurs font appel à un champ lexical relevant d'une réalité technologique concrète ainsi qu'à un utilisateur interagissant par le biais d'un dispositif technique. Ceci semble favoriser une vision du concept d'algorithme en tant que texte d'une procédure exécutable sur une certaine « machine ». Ceci nous semble davantage correspondre au concept de programme informatique qu'à celui d'algorithme. Modeste (2012) décrit ce phénomène consistant à faire écrire hors d'un dispositif d'exécution des

²⁴ « Ressources pour le lycée », *op. cit.* (note 14).

²⁵ *Ibid.*, p. 3.

²⁶ *Op. cit.*, p. 6 (note 6).

²⁷ *Ibidem.*

algorithmes contenant des instructions fortement liées à une machine, et nommés ces produits de la transposition didactique « programmes-papier », considérés comme caractéristiques d'un écrasement des notions d'algorithme et de programme.

- *Problèmes et algorithmes*

On peut noter dans le discours des documents officiels de 2009-2012 que les compétences associées à la résolution de problème ne sont pas spécifiquement de nature algorithmique. En réalité, le terme « problème » utilisé est à entendre dans un sens général, et non dans le sens particulier de *problème algorithmique*, qui reste lui non-défini. L'algorithme est donc entendu non pas (ou pas uniquement) comme la solution d'un problème algorithmique, mais comme outil mobilisable dans l'étude expérimentale d'un problème mathématique ou scientifique. Ceci correspond plutôt au rôle de la programmation dans une démarche expérimentale ou d'investigation, au sens où c'est bien l'exécution d'instructions et de calculs (impossibles à la main) par une machine programmée qui permet de générer des exemples ou des contre-exemples, de simuler des modèles ou de tester des conjectures, et non l'algorithme en lui-même en tant que méthode. Le rapport du CREM (Kahane, 2002) contrairement aux programmes, identifie bien cette distinction.

De plus, la notion d'instance ou de paramètres d'un problème n'apparaît qu'en filigrane (par exemple de manière

implicite dans la description de l'étape de préparation des données en vue du traitement) comme l'ensemble des données initialement disponibles ou saisies par « l'utilisateur ». De même, la notion formelle de résultat d'un algorithme peut être vue comme identifiée à la suite intégrale des effets, « sorties » ou autres manifestations observables effectuées par l'algorithme (ou plutôt par le programme qui l'implémente).

À ce titre, le document « Évolutions de l'écriture des algorithmes au bac » de 2017 (*supra*, note 16) propose des évolutions assez caractéristiques : tout en semblant s'éloigner d'un langage de programmation par une simplification de la syntaxe et la suppression des instructions « saisir » et « afficher », elles suppriment la structuration en entrées-sortie qui caractérisent l'algorithme, et proposent ainsi un point de vue où, suite à un certain nombre de commandes, les variables d'un environnement sont modifiées, sans distinctions entre elles (effets de bord). Ceci montre que subsiste une difficulté à distinguer le concept d'algorithme de celui de programme dans le discours de l'institution.

- *Présentation et représentation des algorithmes*

Le niveau de description des algorithmes proposé dans les programmes de lycée de la réforme de 2009-2012 est « *une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel* ».

La question des niveaux de description n'est ici pas complètement claire. Le programme annonce tout d'abord proposer une « *formalisation en langage naturel* » (expression potentiellement contradictoire) avant d'évoquer la possibilité d'un « *langage symbolique* », sans plus de précision. Par ailleurs, la notion de langage de programmation est évoquée par le biais des calculatrices et logiciels.

Les programmes explicitent le fait qu'aucun langage n'est imposé, ce qui semble aller à l'encontre de l'idée de formalisation. Se pose également la question de l'évaluation des apprentissages des élèves, notamment dans le contexte du baccalauréat : comment par exemple évaluer la capacité des élèves à « *interpréter des algorithmes* » s'il n'existe pas de langage commun à tous les élèves pour les écrire, ni de sémantique pour en définir l'interprétation attendue ?

Le choix d'un pseudo-langage relativement normalisé dans les documents institutionnels ne garantit en aucun cas la cohérence de la présentation des algorithmes²⁸. En l'absence d'une description claire de la sémantique des conventions utilisées (particulièrement en ce qui concerne le statut des variables), l'usage d'un tel niveau de description est même susceptible d'introduire un grand nombre d'implicites potentiellement sources de difficultés.

²⁸ L'annexe A présente une analyse détaillée du pseudo-langage utilisé dans le document-ressource accompagnant les programmes de seconde de 2009 (cf. ressources 2009 du MEN, note 6).

Sur le plan de la représentation des algorithmes le document « Évolutions de l'écriture des algorithmes au bac » de 2017 est aussi éclairant sur la difficulté des *curriculums* à se positionner. On y trouve :

- des modifications syntaxiques sans changements majeurs quant à la sémantique (comme l'introduction de « \leftarrow ») ;
- la conservation de convention des instructions « Fin... » sans utilité sémantique en présence d'indentation ;
- des changements présentés comme simplement syntaxiques alors que leur portée sémantique est d'envergure, comme la suppression de la déclaration des entrées-sorties (avec mention des variables et des hypothèses les concernant renvoyées aux énoncés), ce qui revient à une indifférenciation *a priori* des statuts fonctionnels des variables de l'algorithme.

Conclusion de la section

Entre les périodes 2009-2012 et 2017-2019, les orientations des programmes d'enseignement et leur transposition didactique des concepts liés à l'algorithmique et la programmation ont connu des changements rapides et radicaux. Après un amalgame entre algorithmes et programmes dans les textes de la période 2009-2012 et dans les ressources et manuels (Modeste, 2012), les aménagements curriculaires de 2017 ont

choisi comme référence une définition de l'algorithme prenant davantage en compte la notion de « problème algorithmique », et en particulier sa dimension fonctionnelle : à toute instance du problème qu'il résout, l'algorithme associe un résultat correspondant.

Après une relative invisibilité, volontaire ou fortuite, de la programmation et de certains aspects techniques qui y sont liés (syntaxe, environnement matériel, etc.), celle-ci fait désormais partie des contenus visés et constitue un aspect explicite d'un thème à part entière des programmes.

Après une absence de normalisation de la présentation des algorithmes et un recours à une diversité de langages et outils de programmation *ad hoc*, les enseignements de mathématiques de lycée s'appuient aujourd'hui sur un langage de référence, Python, de notoriété internationale et dont l'usage dépasse le contexte de l'enseignement.

Les évolutions du discours institutionnel sur le savoir à enseigner montrent une prise en compte progressive de certains enjeux épistémologiques et didactique relatifs à l'algorithmique, cependant on note qu'à la fin de la période étudiée subsistent encore un certain nombre de points de tension épistémologiques.

Nous verrons dans la suite de l'article en quoi le logiciel *AlgoBox* est un témoin de premier plan de ces changements successifs, depuis son appa-

rition en 2009 à la suite de la réforme, sa large adoption par les manuels et les enseignants dans les années qui ont suivi, jusqu'à sa quasi-disparition à partir de 2017 au profit du langage Python.

Analyse du logiciel *AlgoBox*

Après cette analyse de la transposition didactique en jeu au lycée, nous nous intéressons au logiciel *AlgoBox* conçu et largement utilisé lors de la période 2009-2019. Après une rapide présentation générale de l'interface du logiciel et de son utilisation et un historique de son évolution, nous décrivons en détail quelques-uns de ses choix de conception (apparents ou revendiqués), ce qui nous permettra d'en tirer une analyse de la transposition informatique du concept d'algorithme à l'œuvre dans ce logiciel.

Présentation générale

Le logiciel libre *AlgoBox* a été conçu et développé à partir de 2014, de manière essentiellement autonome, par Pascal Brachet, professeur de mathématiques au lycée. Il est également l'auteur de nombreuses fiches pédagogiques distribuées librement en ligne, et de plusieurs autres logiciels libres, dont l'éditeur LaTeX *Texmaker*²⁹.

²⁹ Toutes ces ressources sont accessibles sur le site de Pascal Brachet [URL : <https://www.xmlmath.net>].

Aperçu d'utilisation

Le principe du logiciel *AlgoBox* est de permettre la création du texte d'un algorithme, rédigé dans une syntaxe évoquant un pseudo-langage algorithmique de type impératif très proche des recommandations du document-ressource³⁰.

Le logiciel est constitué d'un environnement de travail (interface graphique contenant menus, boutons, zones de rétroaction textuelles et graphiques) et d'un langage textuel permettant l'écriture d'algorithmes, que nous surnommerons par commodité « langage *AlgoBox* ». En termes d'apparence et d'ergonomie, l'interface est proche de celle d'un environnement de développement intégré classique (*Integrated Development Environment* ou IDE), dont elle reprend certaines conventions : coloration syntaxique, repliement de portées, etc.

Le logiciel propose deux modes de rédaction d'algorithmes. Le premier, que nous qualifierons de « mode classique » (cf. Fig. 1), fonctionne à l'aide d'une série de boutons. Chaque appui ouvre une fenêtre de dialogue permettant de spécifier les divers paramètres d'une nouvelle instruction, qui est ensuite insérée à la ligne courante. L'utilisateur ne saisit donc pas intégralement le texte de l'instruction. Un bouton est grisé lorsque la ligne courante n'autorise pas l'insertion d'une instruction de ce type.

Le second mode de rédaction ou « mode éditeur », permet de saisir directement le texte de l'algorithme, et propose les outils classiques d'un environnement de programmation moderne (complétion automatique de mots-clés, aide à la composition). Des boutons permettent de vérifier la conformité syntaxique de l'algorithme ou de basculer vers le mode classique.

Une fois un algorithme composé, il est possible de l'exécuter, c'est-à-dire d'en observer concrètement les effets sur la machine, en cliquant sur le bouton « Tester Algorithme ». Ceci provoque l'ouverture d'une nouvelle fenêtre (cf. Fig. 2) où est repris le texte de l'algorithme, et où apparaissent progressivement les diverses invites de saisie et affichages provoqués par son exécution. Des outils permettent de contrôler le processus, par exemple en procédant à une exécution pas à pas permettant de visualiser les valeurs successives des variables.

Choix de design et philosophie

AlgoBox se présente³¹ comme un logiciel « gratuit, libre, multi-plateforme et facile d'utilisation » et « 100 % conforme aux programmes du lycée et du collège » pour l'initiation à l'algorithmique dans le contexte des programmes de mathématiques de collège et de lycée. Il met

³⁰ MEN, « Programme d'enseignement de mathématiques... », 2009, *op. cit.*, note 6.

³¹ Toutes les citations de cette section sont issues du site *AlgoBox* [URL : <https://www.xmlmath.net/algobox/>].

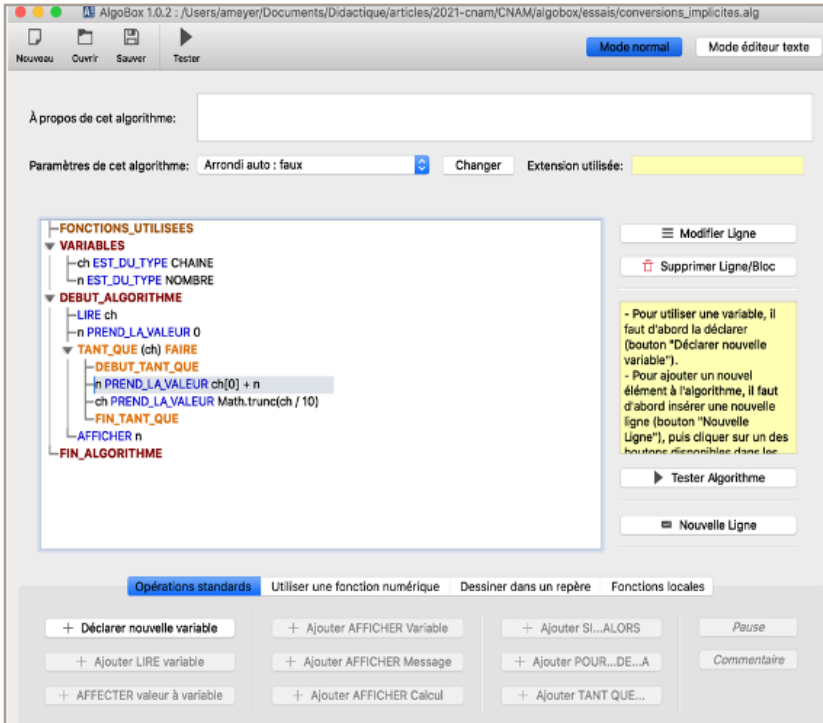


Figure 1 - Fenêtre principale d'AlgoBox (mode classique)

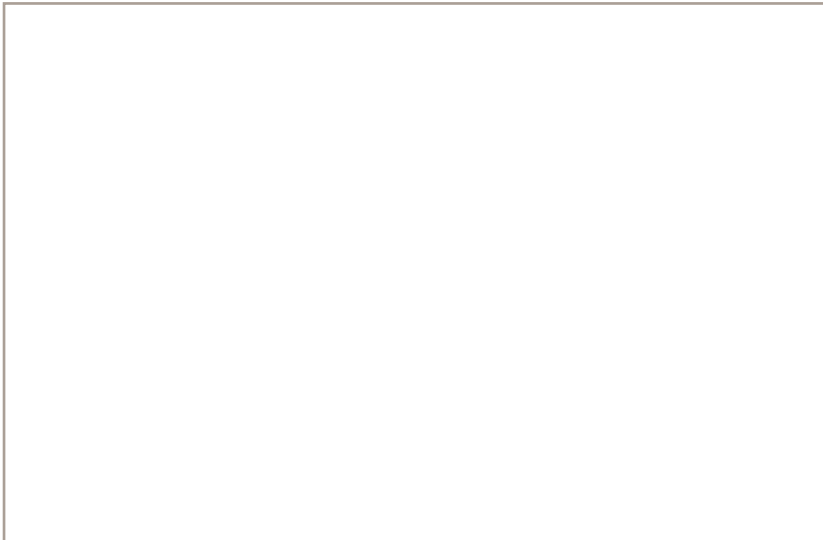


Figure 2 - Fenêtre d'exécution

en avant un certain nombre de caractéristiques, dont les suivantes :

- « *Un apprentissage centré sur la logique, pas sur les problèmes de syntaxe* » ;
- « *Une interface en français, moderne et ergonomique* » ;
- « *La syntaxe la plus proche de celle des exercices au Bac* » ;
- « *Une approche réaliste, pédagogique et non élitiste de l'algorithmique* ».

À la lecture de ces arguments, il apparaît que la vocation principale de ce logiciel est de répondre aux besoins introduits par la réforme de 2009. Il est également clair que la philosophie de ce logiciel est de permettre aux enseignants de mathématiques et à leurs élèves de travailler sur les notions algorithmiques en faisant l'économie de l'apprentissage d'une syntaxe nouvelle. Il est ici sous-entendu que la syntaxe du langage *AlgoBox* « va de soi », du fait d'être si « *proche du langage naturel algorithmique* » (à supposer qu'un tel langage existe et soit unique).

Comme on le détaillera par la suite, on peut aussi supposer à la lecture des divers documents relatifs à ce logiciel que les algorithmes composés sous *AlgoBox* se veulent idéaux, débarrassés des contingences et tracasseries associées (à tort ou à raison) à l'activité de programmation informatique : nécessité de familiarisation avec des procédures techniques inhé-

rituelles, prise en compte des contraintes ou limites inhérentes à une architecture matérielle, à une modélisation et représentation informatique des données, etc. Une pratique trop technique ou technologique de la programmation apparaîtrait ainsi comme « *élitiste* ». *AlgoBox* se veut donc un logiciel « *démocratique* ».

La lecture des recueils d'activités publiés par l'auteur lui-même pour les classes de première et terminale permet de nuancer ces revendications d'« évidence » : une annexe fournie, d'une dizaine de pages, y présente les « *équivalences entre pseudo-codes* », « *problèmes de syntaxe* » les plus fréquents et astuces de fonctionnement d'*AlgoBox*³².

Ce logiciel fournit cependant plus qu'une simple interface de rédaction « *pas à pas [...] hiérarchique et structurée* » d'algorithmes. La grande force revendiquée par ce logiciel est de permettre à la fois la composition d'algorithmes et la capacité à les faire fonctionner (les « *tester* » ou les exécuter) comme on ferait fonctionner des programmes. Il s'agit donc d'une véritable tentative de transposition informatique du concept d'algorithme. Nous montrerons que cette ambition se heurte à des difficultés d'ordres épistémologique et didactique.

³² Voir les brochures rédigées par P. Brachet : « Algorithmique en classe de première avec *AlgoBox* » [URL : <https://www.xmlmath.net/algobox/algobook.html>] ; « Algorithmique en classe de terminale avec *AlgoBox* » [URL : <https://www.xmlmath.net/algobox/algobook.html>].

Afin de caractériser la conception de l’algorithmique qui sous-tend le logiciel *AlgoBox*, le reste de cette section est consacré à un rappel chronologique de l’apparition de ses différentes fonctionnalités en regard avec l’évolution des programmes, et à une étude plus précise des caractéristiques techniques d’*AlgoBox*.

Historique des versions

La première version d’*AlgoBox* (version 0.1) a été rendue publique au début du mois de juillet 2009³³, quelques mois après la publication des projets de nouveaux programmes de mathématiques de seconde et la consultation qui l’a accompagnée³⁴. Elle a également suivi de peu la publication du premier document d’accompagnement et des programmes définitifs³⁵. Cette version fournissait déjà la plupart des fonctionnalités de base du logiciel, mais elle a depuis connu un certain nombre d’évolutions.

En quelques mois, *AlgoBox* a connu plusieurs mises à jour : sortie des versions

0.2 et 0.3 l’été 2009, puis 0.4 en octobre 2009. Le rythme s’est ensuite stabilisé à une nouvelle version par an environ, jusqu’à la version 0.9 en 2014. Enfin, après une pause de trois ans, la version 1.0 d’*AlgoBox* est apparue à l’été 2017, elle n’a connu que des mises à jour techniques mineures depuis³⁶.

Nous soulignons ici quelques évolutions représentatives des liens entre les évolutions curriculaires et les versions d’*AlgoBox* : l’apparition en 2010 (version 0.5) et 2011 (version 0.6) dans le logiciel de nouvelles fonctions mathématiques prédéfinies (calculs statistiques sur les listes en 2010, lois et distributions en 2011) coïncidant avec l’introduction de l’algorithmique dans les programmes de première et de terminale, incluant ces notions de statistiques et de probabilités ; et l’apparition des fonctions dites « locales » en 2017 (version 1.0) de façon concomitante à l’adaptation des programmes de seconde suite à la réforme du collège, et l’introduction de la notion de fonction informatique.

Caractéristiques techniques

Nous présentons ici une synthèse des caractéristiques techniques d’*AlgoBox* qui témoignent de choix spécifiques

³³ Tous ces éléments factuels sont issus de la page « Changelog » du site *AlgoBox* [URL : <https://www.xmlmath.net/algobox/changelog.html>], ainsi que des archives de code source disponibles sur la page [URL : <https://www.xmlmath.net/algobox/download.html>].

³⁴ DGESCO, « Consultation. Projet de programme de mathématiques. Classe de seconde générale et technologique », mars 2009. [URL : https://web.archive.org/web/20090319032830/http://eduscol.education.fr/D0015/consult_Maths.htm].

³⁵ « Programme d’enseignement de mathématiques... » et « Ressources pour la classe de seconde – Algorithmique », 2019, *op. cit.*, note 6.

³⁶ Nous détaillons dans l’annexe B une sélection de modifications apportées au logiciel, en nous focalisant sur les évolutions liées à la syntaxe et à la sémantique du langage, et en les mettant en regard certains éléments de l’évolution des *curricula*, lorsque cela nous semble éclairer celle du logiciel.

ou d'une certaine prise de distance avec le savoir savant³⁷.

Le langage *AlgoBox* se veut lisible et compréhensible par tout élève, sans apprentissage particulier de sa syntaxe. Il emploie donc essentiellement des mots-clés en français. Par exemple, le terme « prend la valeur », choisi pour l'affectation, est conforme à l'expression utilisée par le document d'accompagnement de 2009³⁸. Dans l'ensemble, le langage *AlgoBox* est effectivement très proche de ce pseudo-langage officiel.

Une fois un algorithme décrit, il est possible de l'exécuter afin d'en observer le fonctionnement. Ceci est rendu possible par la traduction ligne à ligne de l'algorithme en un programme JavaScript, puis son exécution au sein d'une page web affichée dans la fenêtre d'exécution (cf. Figure 2). Le langage *AlgoBox* s'apparente donc à une surcouche fine de JavaScript plutôt qu'à un langage de programmation à part entière.

Un algorithme est constitué dans *AlgoBox* de plusieurs rubriques : un bloc *FONCTIONS_UTILISEES* (depuis la version 1.0), un bloc *VARIABLES*, et le corps de l'algorithme lui-même, délimité par les mots-clés *DEBUT_ALGORITHME* et *FIN_ALGORITHME*. De même, tous les blocs (de boucles,

de conditionnelles) sont délimités par des marqueurs explicites *DEBUT_...* et *FIN_...*. Dans le mode de travail « classique », *AlgoBox* représente l'imbrication des blocs et instructions sous une forme arborescente et indentée, proche de la présentation des algorithmes dans le document-ressource de 2009 (MEN, 2009a), comme on le voit en Figure 1. Sur le plan technique, cette forme arborescente pourrait se passer de délimiteurs explicites, ce qui permettrait d'alléger la syntaxe.

Depuis sa version 1.0, *AlgoBox* permet à l'utilisateur de définir des fonctions dites « locales », qui peuvent ensuite être utilisées dans l'algorithme principal. Même si cette fonctionnalité permet effectivement de manipuler des fonctions personnalisées (principalement numériques), elle reste à un stade de développement très incomplet.

Dans la description d'un algorithme, le bloc *VARIABLES* recense toutes les variables de l'algorithme et définit leur type. Une telle contrainte n'existe pas dans le langage-cible JavaScript, dans lequel une variable peut être introduite en tout point du programme, et sans en spécifier le type (JavaScript obéissant à un typage dynamique). Cette différence de philosophie entraîne un certain nombre d'effets indésirables.

AlgoBox permet en principe de manipuler trois types de valeurs : nombres, listes de nombres et chaînes de caractères. Ceci implique en particulier qu'il n'existe

³⁷ Ces caractéristiques sont analysées plus en détail dans l'annexe C (Caractéristiques techniques détaillées).

³⁸ « Programme d'enseignement de mathématiques... », *ibid.*

pas de type « nombre entier », ce qui peut sembler étonnant dans ce contexte.

Ainsi, *AlgoBox* présente certaines caractéristiques potentiellement problématiques. Malgré une réelle proximité avec l'esprit du pseudo-langage employé dans le document-ressource³⁹, le langage *AlgoBox* est entaché d'une certaine hétérogénéité (de langue, de style et de sémantique). Sa syntaxe n'est pas entièrement circonscrite, certaines instructions pouvant faire appel à du code JavaScript arbitraire. Un certain nombre d'effets indésirables relatifs à une insuffisance du contrôle de type sont également visibles. Nous y voyons les conséquences non maîtrisées d'une traduction imparfaite des algorithmes vers Javascript.

Selon nous, l'absence de spécification formelle de la syntaxe, le non-emploi de techniques de compilation appropriées pour la traduction des algorithmes, et le choix de déléguer l'analyse et l'exécution du code à un interpréteur JavaScript obéissant à ses propres contraintes techniques, produisent certains comportements à première vue incompréhensibles. Ces aspects d'*AlgoBox* sont susceptibles d'entraîner des difficultés conséquentes chez certains utilisateurs, particulièrement les débutants auxquels s'adresse en priorité ce logiciel.

Ces observations nous confortent dans l'idée que le langage et le mécanisme d'exécution d'*AlgoBox* souffrent

d'une importante imprécision et d'un caractère hétérogène, voire imprévisible, et remet en question le projet d'*AlgoBox* de se présenter comme libéré des difficultés techniques supposées d'un « véritable » langage de programmation.

À un moment de l'histoire de l'enseignement de l'informatique et à une place du cursus où ces activités algorithmiques et de programmation étaient entièrement nouvelles pour un grand nombre d'élèves et d'enseignants, il est légitime de s'interroger sur l'impact qu'un tel outil a pu avoir sur les conceptions de ses utilisateurs (élèves ou enseignants). Dans ce sens, nous allons nous intéresser à la transposition informatique du concept algorithme réalisée par le logiciel *AlgoBox*.

Analyse de la transposition informatique

Le spectre thématique visé par *AlgoBox* est très nettement ancré dans le programme de mathématiques du lycée, comme en témoigne la palette de ses fonctionnalités (fonctions prédéfinies, repère graphique, etc.) ainsi que toute la documentation qui l'accompagne. Comme nous l'avons indiqué précédemment, *AlgoBox* ne se présente pas comme un environnement d'apprentissage de la programmation mais comme un logiciel « *d'aide à l'élaboration et à l'exécution d'algorithmes dans l'esprit des nouveaux programmes de mathématiques du secondaire* », proposant « un

³⁹ *Ibidem*.

apprentissage centré sur la logique, pas sur les problèmes de syntaxe ». Cette déclaration d'intention nous semble relever d'un projet de transposition informatique du concept d'algorithme (et donc, dans une certaine mesure, d'une transposition didactique), que nous nous proposons d'analyser.

- *Absence de repérage du statut fonctionnel des variables*

Il n'existe aucune distinction *a priori* dans *AlgoBox* entre les différents statuts fonctionnels des variables évoqués au paragraphe « Repères épistémologiques ». Toutes les variables sont déclarées dans la rubrique VARIABLES. Il est donc impossible de déterminer lesquelles sont utilisées pour représenter « l'entrée », ou instance courante du problème (*variables-paramètres*), lesquelles, si elles existent, servent à construire le résultat final (*variables-résultat*), et lesquelles sont de simples variables de travail. En réalité, aucun mécanisme autre que les instructions de saisie et d'affichage ne permet à un algorithme *AlgoBox* de spécifier des paramètres de l'algorithme ou de renvoyer un résultat.

Une reconstruction *a posteriori* de ces catégories pourrait donc consister à considérer les variables qui font l'objet d'une saisie comme des variables-paramètres, celles qui font l'objet d'un affichage comme des variables-résultats, et les autres comme des variables de travail. Malheureusement ce critère se révèle insuffisant, notamment lorsque des affichages ou

des saisies sont effectués pour de simples raisons « cosmétiques » ou d'ergonomie (affichage et choix dans un menu, invitations à « arrêter ou continuer », etc.), qui ne répondent pas à la *spécification* du problème algorithmique posé.

Cette impossibilité d'identification claire des paramètres et résultats entretient une confusion entre la spécification d'un algorithme et l'interface utilisateur d'un « logiciel ». Ceci nous semble problématique, du fait que la distinction entre les différents statuts fonctionnels des variables et la compréhension des différents rôles joués par des saisies et affichages sont des points notoirement délicats de l'enseignement de la programmation aux débutants.

- *Confusion algorithme-programme*

Un autre axe d'analyse de la transposition informatique à l'œuvre dans *AlgoBox* concerne la distinction (ou la confusion) entre les concepts d'algorithme et de programme. Le logiciel, dans son ambition de libérer l'élève débutant des difficultés liées à l'apprentissage de la syntaxe d'un langage de programmation, présente donc le paradoxe de vouloir proposer un langage à la fois « évident » et exécutable sur machine. Or, il nous semble possible que cette ambition soit d'emblée vouée à l'échec, car comme nous l'avons vu, les concepts d'algorithme et de programme sont distincts et possèdent leurs caractéristiques propres.

Comme nous l'avons rappelé dans la section consacrée à l'analyse des programmes, et conformément à l'étude approfondie menée dans (Modeste, 2012, chap. 6), les programmes de la période 2009-2012 font apparaître un important amalgame entre ces deux concepts. Cette étude y décèle aussi une tentative de résumer l'activité algorithmique à un pur travail sur le langage, détaché d'une forme d'activité intellectuelle spécifique (il suffirait en quelque sorte de savoir lire ou écrire avec rigueur pour que « tout fonctionne »).

AlgoBox propose à ses utilisateurs de rédiger des algorithmes, et non des programmes. Ces algorithmes font cependant explicitement référence à un utilisateur par le biais des instructions de saisie et d'affichage, et donc, par extension, à une machine, même si ces notions relèvent du concept de programme. Le logiciel permet ensuite d'exécuter ces algorithmes, sans traduction visible vers un langage de programmation identifié comme tel, et en tentant de « gommer » les éventuelles difficultés techniques ou syntaxiques inhérentes à la programmation (avec plus ou moins de succès, comme on l'a vu).

Il apparaît donc qu'*AlgoBox* vise une transposition informatique conforme au concept d'algorithme tel qu'il transparaît dans les programmes de 2009-2012. Ce faisant, il tente de concrétiser l'« écrasement conceptuel » entre algorithmes et programmes que nous avons décrit, escamotant le concept de programme infor-

matique à l'intérieur même d'un langage de programmation.

- *Concept d'algorithme et pensée informatique*

Comme discuté ci-dessus, il n'existe pas dans *AlgoBox* de mécanisme permettant d'identifier les paramètres ni le résultat d'un algorithme. Par conséquent, il est impossible d'y concevoir un algorithme faisant appel à un autre algorithme réalisé précédemment, autrement qu'en le recopiant intégralement. Pour l'enseignant, cela signifie que tout problème doit être résolu « en une fois », et que sa solution ne pourra plus resservir (sauf à être reprise intégralement et adaptée).

Pour des raisons semblables, l'absence dans la transposition informatique opérée par *AlgoBox* des concepts de paramètres et de résultat fait obstacle au développement d'une approche *généralisatrice*, qui consiste à résoudre un problème spécifique en l'élargissant à un problème plus général, par exemple en introduisant de nouveaux paramètres, ou tout simplement à résoudre un problème plus général en modifiant un algorithme déjà produit pour un problème particulier.

Un tel travail est certes envisageable, par exemple en remplaçant l'affectation d'une constante à une variable par une saisie. Cependant, nous défendons l'idée qu'une formalisation plus rigoureuse du concept de paramètre est indispensable pour aborder de manière satisfaisante ce type de démarche. L'impact possible d'un

tel manque de conceptualisation est également illustré par ce que Modeste (2012) appelle « algorithmes-instanciés » et qu'il identifie comme symptomatiques de la période 2009-2012.

Comme on le voit, les caractéristiques d'*AlgoBox* ne permettent pas, ou pas pleinement, de mettre en œuvre certains types de démarches qui, comme nous l'avons déjà mentionné dans la partie « Repères épistémologiques », sont couramment considérés comme constitutifs de la pensée informatique.

- *Conclusion sur la transposition informatique*

Selon tous les critères évoqués dans cette section, liés au statut des variables, à la syntaxe et à la sémantique du langage, à la confusion entre algorithmes et programmes, au concept de problème (embarquant ceux de paramètre et de résultat) et à une certaine définition académique de la pensée informatique, on peut donc affirmer que la transposition informatique à l'œuvre dans *AlgoBox* obéit à un parti pris relativement éloigné du savoir savant.

Étant donné la large adoption de cet outil à partir de 2009, ces choix didactiques et épistémologiques (délibérés ou non) sont donc susceptibles d'avoir eu un impact sur les apprentissages et sur le type de méthodes et de raisonnements accessibles aux enseignants et aux élèves qui l'ont utilisé. Dans la dernière section, nous questionnerons la place et l'influence d'*AlgoBox* vis-à-vis de ce *curriculum*.

Place et influence d'*AlgoBox* au sein du *curriculum*

Dans cette section, nous souhaitons mettre en lumière les phénomènes qui lient le déploiement du logiciel *AlgoBox* et celui du *curriculum* de mathématiques de lycée entre 2009 et 2019 (tant du point de vue du savoir à enseigner que du savoir enseigné).

Un premier pas vers l'algorithmique en mathématiques

En 2009, l'enseignement de l'informatique est essentiellement absent de l'enseignement secondaire public général en France, à l'exception d'une part d'algorithmique dans les mathématiques de la filière littéraire, malgré des efforts de plusieurs organismes et groupements d'intérêts (sociétés savantes ou syndicats professionnels, Académie des sciences, etc.) de le promouvoir.

En lien avec les recommandations de la « commission Kahane » (Kahane, 2002), la réforme du lycée de 2009-2012 généralise l'introduction de l'algorithmique dans les mathématiques du lycée général. Elle présente la démarche algorithmique comme étant « *depuis les origines, une composante essentielle de l'activité mathématique* »⁴⁰, et par conséquent fortement ancrée dans cette disci-

40 « Ressources pour la classe de seconde – Algorithmique », 2019, *op. cit.*, note 6.

pline. Elle en confie la prise en charge aux enseignants de mathématiques, sans déployer de véritable formation spécifique.

Ceci, ainsi que les conditions et le rythme de la réforme, associés à l'absence de langage imposé et à la consigne de transversalité (interdisant de fait la tenue de séances spécifiques) décourage l'adoption par les enseignants de langages de programmation « complets », dont la mise en place en classe est souvent perçue comme inaccessible. Ceci est renforcé par la perception par certains enseignants du caractère exogène de ces contenus, souvent absents de leur formation initiale.

L'apparition d'*AlgoBox* à la rentrée 2009 vient ainsi combler un besoin : un outil facile à prendre en main, présenté comme parfaitement adapté et focalisé sur le contenu du programme, ne nécessitant pas l'apprentissage d'une syntaxe particulière, permettant de réaliser les activités algorithmiques attendues des programmes, notamment dans une démarche de résolution de problème.

Une transposition informatique fidèle aux instructions officielles

Comme nous l'avons détaillé dans la section précédente, une analyse soignée de la transposition informatique du concept d'algorithme à l'œuvre dans *AlgoBox* révèle une grande conformité avec les consignes et préconisations des programmes. Cette fidélité est pleine-

ment revendiquée par l'auteur d'*AlgoBox*. Notre analyse a aussi révélé un certain nombre de particularités dont il est légitime de questionner les effets.

En refusant en un sens de prendre en charge les aspects spécifiques et potentiellement complexes de l'informatique et de la programmation tels que l'apprentissage d'une syntaxe rigoureuse, les enjeux de représentation des données et en particulier des nombres, et plus généralement de modélisation informatique des objets manipulés, le logiciel semble entretenir l'idée (ou l'illusion) qu'il « suffit de savoir lire » pour savoir programmer.

Paradoxalement, et en contradiction avec sa propre philosophie, en imposant un mode de fonctionnement spécifique (types d'objets restreints, présentation normalisée, vocabulaire fixé), il est dans le même temps susceptible d'aller à l'encontre des avantages offerts par une expression plus libre des algorithmes telle qu'on la rencontre couramment dans nombre de cursus universitaires en informatique et en mathématiques, et même dans le document-ressource⁴¹.

Enfin, en ne permettant pas (ou difficilement) l'émergence d'un raisonnement par décomposition ou par généralisation, ni le réemploi d'algorithmes précédemment étudiés, il fait obstacle à la mise en œuvre de modes de raisonnement pourtant très répandus en informatique

⁴¹« Programme d'enseignement de mathématiques... », 2019, *op. cit.*, note 6.

comme en mathématiques, en particulier dans le contexte d'une « *démarche de résolution de problème* » fréquemment citée comme objectif d'apprentissage dans les programmes.

Ainsi, *AlgoBox* produit une transposition informatique du concept d'algorithme fidèle aux programmes de 2009-2012. Par nécessité, le logiciel formalise certains éléments implicites des instructions officielles, révélant ainsi un certain nombre de phénomènes de transposition didactique qui semblent n'avoir pas été anticipés par les auteurs des programmes, et dont nous avons montré un certain nombre de limites ou d'aspects problématiques.

Un projet voué à l'échec ?

Les sections précédentes ont montré comment les programmes de mathématiques du lycée de 2009-2012 en algorithmique sont porteurs d'un projet (qui reste implicite, et peut-être non-intentionnel) : celui d'un langage d'algorithme dont la syntaxe et la sémantique seraient transparentes, et dont l'implémentation en machine devrait éviter tout aspect technique. Le logiciel peut alors être vu comme une tentative de réalisation de ce projet.

Selon nous, ce projet est intrinsèquement porteur de contradictions qui condamnent toute tentative de réalisation à se heurter à des difficultés d'ordres épistémologique, technique et didactique.

Sur le plan épistémologique, l'écrasement des notions d'algorithme et de programme que nous avons mis en lumière vient se heurter à la nature fondamentalement différente de ces deux concepts :

- le programme, comme texte formel avec des règles syntaxiques rigoureuses et doté d'une sémantique liée à une machine (réelle ou virtuelle) ;
- l'algorithme, comme expression dans un langage partiellement formalisé d'une méthode de résolution de problème, indépendamment de certaines contraintes techniques.

Il en découle un langage, dans le cas qui nous occupe le langage *AlgoBox*, dont on pourrait dire qu'il vise à rendre les algorithmes-papier (au sens de Modeste, 2012) exécutables. On peut dire qu'il cherche à invisibiliser les aspects techniques et informatiques (notamment en éludant la question du modèle de calcul ou de la machine, ou par ses tergiversations quant à la manipulation des nombres). Nous avons montré un certain nombre de problèmes techniques du logiciel et de son langage qui nous semblent révélateurs des limites de cette tentative d'invisibilisation.

Enfin, on peut voir derrière le discours de simplicité et d'accessibilité d'*AlgoBox*, une intention pédagogique résumée ainsi sur son site : « *un logiciel éducatif basé sur une logique pédagogique (apprentissage de l'algorithmique*

par structures logiques à travers un langage textuel proche du langage naturel algorithmique) » (souligné par nous). De ceci découle une tentative d'évitement des deux stratégies d'initiation à l'algorithmique identifiées par Nguyen (2005) : présence d'une machine de référence, ou appui sur un langage représentatif d'une classe de langages de programmation. Sur le plan didactique, cette position ne semble pas tenable.

Ainsi, les programmes de 2009-2012 et le logiciel, dans une illusion de transparence (Chevallard, 1985), cherchent à proposer une copie simplifiée, mais fidèle du concept algorithme (pour reprendre la formule d'Artigue, 1990), et se heurtent alors à des problèmes de transposition didactique et informatique.

Les aménagements des programmes survenus en 2017, et plus encore la réforme du lycée de 2019-2020, distinguent plus nettement les notions d'algorithme et de programme et désignent Python comme langage de référence pour la programmation. Ces changements ont donné un coup d'arrêt à l'utilisation d'*AlgoBox*. En témoignent le document-ressource⁴², dont les exemples sont exclusivement rédigés en Python, l'absence de mise à jour significative d'*AlgoBox* depuis 2017, son retrait de la liste des logiciels fournis au CAPES de mathématiques dès la session 2018, et la disparition des mentions de ce logiciel dans les manuels scolaires et canaux de diffusion habituels (dont le site *éduscol*).

Ce tournant s'est vu renforcé par l'apparition dès 2019 d'une matière de tronc commun en classe de seconde (SNT) et d'une spécialité du cycle terminal (NSI) pleinement consacrées à l'informatique et mettant un fort accent sur l'apprentissage de la programmation, de la création d'un nouveau corps de professeurs pour enseigner ces matières (CAPES NSI).

Une influence forte sur le savoir enseigné

Selon l'auteur d'*AlgoBox*, « *la quasi-totalité des manuels scolaires de mathématiques (de la seconde à la terminale)* » citent ou utilisent ce logiciel. Nous avons pu observer qu'une grande majorité des ressources et la quasi-totalité des manuels de la période 2009-2017 présentent effectivement leurs algorithmes avec son langage (éventuellement parmi d'autres langages).

L'une des raisons de cette diffusion large et rapide dans les manuels, les ressources et donc les classes est certainement la forte proximité d'*AlgoBox* avec le contenu des programmes. Les contraintes qui se sont exercées sur la mise en œuvre de ces programmes, mentionnées plus haut (faible formation des enseignants, nature et urgence de la mise en place de la réforme...), ont aussi probablement joué un rôle.

En effet, la transposition didactique du concept algorithme proposée par les

⁴² Documents-ressources MEN 2017, *op. cit.*, note 14.

programmes de 2009-2012 est partagée par les manuels (pour des exemples d'analyses voir Modeste, 2012). On y retrouve aussi dans les ressources et manuels, souvent en cohabitation avec *AlgoBox*, les langages des calculatrices Casio et TI, et les logiciels *Xcas*, *Scratch* et *Scilab*. La proximité entre le type de langage utilisé dans les ressources institutionnelles pour décrire les algorithmes et le langage apparaît alors encore plus nettement. Il est donc très probable que le logiciel ait été utilisé très massivement comment nous le pensons.

Quelques indices laissent penser qu'*AlgoBox* a contribué à modéliser et diffuser un langage de description des algorithmes, inspiré de celui du document-ressource de 2009. Le premier est la présence récurrente dans les sujets du baccalauréat de 2012 à 2019 d'algorithmes basés sur le langage d'algorithmes conforme aux attentes des documents officiels, mais comprenant des corps de blocs délimités par les mots-clés « FinSi », « FinPour » et « FinTantque », caractéristique du logiciel. Ces mots-clés sont conservés dans le document « *évolution de l'écriture des algorithmes au Baccalauréat* » déjà mentionné.

Le deuxième indice est la présence, dans certains manuels ultérieurs à la réforme de 2019, de présentations des algorithmes qui conservent certains éléments de forme et des mots-clés des langages d'*AlgoBox* et du document-ressource de 2009, alors que le langage Python est dorénavant imposé. Il est pro-

bable que chez les enseignants comme chez les rédacteurs de manuels, la période 2009-2019 a contribué à construire des conceptions de la notion d'algorithme et de sa présentation, qui continueront d'influencer le savoir enseigné.

Conclusions

Forquin (2008) identifie un certain nombre de questionnements sociologiques concernant l'objet *curriculum* : en tant qu'opérateur de sélection et de transmission culturelles, en tant que véhicule possible d'une « *culture d'école* » à part entière, et enfin en tant qu'objet à dimension politique. Sur ce dernier point, il écrit :

La question importante est ici celle de savoir qui dispose du pouvoir de contrôle sur l'élaboration (et la promulgation) des programmes et des plans d'étude, et comment ces phénomènes de contrôle se traduisent aux différentes étapes de ce qu'on pourrait appeler « *la chaîne de production curriculaire* » (Forquin, 2008, p. 9).

Il questionne ainsi ce que la théorie anthropologique du didactique appelle la noosphère, et s'intéresse plus précisément aux phénomènes qui entrent en jeu dans la chaîne de production curriculaire. La transposition didactique s'intéresse plus particulièrement aux produits de ces phénomènes.

En étudiant la transposition didactique du concept algorithme dans les

instructions officielles et les ressources d'enseignement de la période 2009-2019, ainsi que dans *AlgoBox*, nous avons mis en lumière de tels phénomènes curriculaires, qui illustrent comment le contrôle dont parle Forquin peut relever d'influences complexes, pouvant dépasser les intentions de ceux qui élaborent les *curriculum*s.

En 2009 paraissent de nouveaux programmes introduisant l'algorithmique dans l'enseignement des mathématiques. Bien que de nombreux logiciels pour l'initiation à l'algorithmique ou la programmation soient déjà disponibles, le logiciel s'impose en proposant un langage parfaitement adapté aux attentes institutionnelles et un logiciel offrant une prise en main facile, dans un contexte où les enseignants manquent de repères. Le logiciel incarne la conception de l'algorithmique portée par les programmes (dont un amalgame algorithme-programme), et son langage a colonisé un certain nombre de ressources pédagogiques, de manuels scolaires et jusqu'aux sujets du baccalauréat.

Une inflexion du discours se produit dans les années 2017-2020, certainement liée aux limites du projet de 2009, au développement de l'informatique comme discipline scolaire, et à diverses prises de position et analyses dans la noosphère. Les programmes de 2019-2020 nous semblent marquer un retour à une situation plus proche d'autres *curriculum*s internationaux, comme en témoigne l'adoption du langage Python.

Il nous a semblé important d'analyser et documenter la période 2009-2019, qui peut être vue comme une sorte de bulle, à la fois une exception française dans le choix d'introduire une part d'algorithmique dans les mathématiques, et dans la transposition didactique qui en découle.

Cette analyse apporte aussi des éléments pour réfléchir à des questions liées à l'enseignement de l'informatique et à son développement comme discipline scolaire : l'impact des choix de logiciels et langages informatiques sur les apprentissages et sur la chaîne de production curriculaire ; la difficulté de conception de logiciels pédagogiques *ad hoc* adaptés à des choix curriculaires précis ; les effets de transposition informatique, y compris pour des concepts issus de l'informatique. En particulier, les choix curriculaires actuels désignent Python comme langage de programmation de référence pour l'enseignement de l'algorithmique et de la programmation au lycée. Ce choix d'un langage de programmation issu du milieu professionnel ne fait pas pour autant disparaître les enjeux de transpositions informatique et didactique.

Bibliographie

Académie des Sciences (2013). *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*. Rapport n° 513. [URL : http://www.academie-sciences.fr/active/rapport/rads_0513.pdf].

Artaud M. (1998). « Introduction à l'approche écologique du didactique – L'écologie des organisations mathématiques et didactiques ». In ARDM (dir.). *Actes de la IX^e École d'été de didactique des mathématiques*. Caen : IUFM de l'académie de Caen, pp. 101-139.

Artigue M. (1990). « Épistémologie et Didactique ». *Recherches en Didactique des Mathématiques*, 10 (2.3), pp. 241-285.

Balacheff N. (1993). « La transposition informatique, un nouveau problème pour la didactique ». In M. Artigue, N. Balacheff, R. Gras, C. Laborde & P. Tavninot (dir.). *Colloque « Vingt ans de didactique des mathématiques en France », 15-17 juin 1993*. Grenoble : La Pensée Sauvage, pp. 364-370.

Baron G.-L. & Bruillard É. (2011). « L'informatique et son enseignement dans l'enseignement scolaire général français : Enjeux de pouvoirs et de savoir ». In J. Lebaume, A. Hasni & I. Harlé (dir.). *Recherches et expertises pour l'enseignement scientifique*. Bruxelles : De Boeck, pp. 79-90.

Bosch M. & Gascón J. (2014). « Introduction to the Anthropological Theory of the Didactic (ATD) ». In A. Bikner-Ahsbahs & S. Prediger (eds.). *Networking of Theories as a Research Practice in Mathematics Education*. Springer International Publishing, pp. 67-83.

Chen C., Haduong P., Brennan K., Sonnert G. & Sadler P. (2019). « The effects of first programming language on college students' computing attitude and achievement: A comparison of graphical and textual languages ». *Computer Science Education*, 29(1), pp. 23-48.

Chevallard Y. (1985). *La transposition didactique – du savoir savant au savoir enseigné*. Grenoble : La Pensée Sauvage.

Chevallard Y. & Bosch M. (2014). « Didactic Transposition in Mathematics Education ». In S. Lerman (ed.). *Encyclopedia of Mathematics Education*. Springer Netherlands, pp. 170-174.

Durand-Guerrier V., Meyer A. & Modeste S. (2019). « Didactical Issues at the Interface of Mathematics and Computer Science ». In G. Hanna, D. A. Reid & M. de Villiers (eds.). *Proof Technology in Mathematics Research and Teaching*. Springer International Publishing, pp. 115-138.

Furquin J.-C. (2008). *Sociologie du Curriculum*. Rennes : Presses Universitaires.

Froidevaux C., Gaudel M.-C. & Soria M. (1990). *Types de données et algorithmes*. Paris : McGraw-Hill.

Gal-Ezer J. & Harel D. (1998). « What (Else) Should CS Educators Know? ». *Communications of the ACM*, 41(9), pp. 77-84.

Kahane J.-P. (2002). *Enseignement des sciences mathématiques : Commission de réflexion sur l'enseignement des mathématiques : Rapport au ministre de l'Éducation nationale*. Paris : Odile Jacob/CNDP.

Modeste S. (2012). « Enseigner l'algorithme pour quoi ? Quelles nouvelles questions pour les mathématiques ? Quels apports pour l'apprentissage de la preuve ? ». Thèse en didactique des mathématiques, Université de Grenoble.

Modeste S. (2020). « Prendre en compte l'épistémologie de l'algorithme. Quels apports d'un modèle de conceptions ? Quelle transposition didactique ? ». *Recherches en Didactique des Mathématiques*, 40(3), pp. 363-404.

Nguyen C. T. (2005). « Étude didactique

de l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement mathématique secondaire à l'aide de la calculatrice ». Thèse en didactique des mathématiques, Université de Grenoble.

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

Selby C. & Woollard J. (2013). *Computational thinking: The developing definition* [Rapport de recherche]. University of Southampton, UK [URL : <https://eprints.soton.ac.uk/356481/>].

Wing J. M. (2006). « Computational thinking ». *Communications of the ACM*, 49(3), pp. 33-35.

Le Goupil G3, point de basculement de l'industrie de la micro-informatique en France

Isabelle Astic

Musée des Arts et Métiers – Le Cnam

La micro-informatique est un tournant dans l'histoire de l'informatique car elle en a permis la diffusion dans tous les secteurs d'activité, qu'ils soient professionnels ou familiaux. L'invention du micro-processeur a permis de réduire la taille de l'ordinateur, d'en faciliter la conception et d'en diminuer le prix de façon considérable, mettant ainsi l'informatique à la portée de nombreuses structures. La conception d'un micro-ordinateur devenait à la portée d'un bon électronicien et des revues comme *Elektor* ou *Micro-Systèmes* ont proposé à leur lecteur de construire le sien. La diffusion du micro-processeur a favorisé l'apparition de constructeurs de micro-informatique en dehors des États-Unis, dès 1973. Trente-cinq d'entre eux existaient en France en 1985 selon l'hebdomadaire *Décision*

*Informatique*¹. SMT, appelé également SMT-Goupil d'après la marque de ses micro-ordinateurs, était l'un d'entre eux.

Le musée des Arts et Métiers conserve un fonds en provenance de cette société et des machines, dont trois Goupil G3², tête de réseau du nanoréseau du plan Informatique pour tous (IPT). Trois G3, n'est-ce pas trop ? D'où proviennent-ils et pourquoi sont-ils entrés au musée ? Que montrent-ils ?

Pour répondre à ces questions, il est nécessaire dans un premier temps de rap-

¹ « Le chagrin et la pitié », *Décisions Informatique*, n° 75, 6 mai 1985, p. 84.

² Pour plus de simplicité, les machines construites par SMT sous l'appellation Goupil Gx, seront simplement nommées Gx dans le reste de cet article.



Figure 1 - Micro-ordinateur Goupil G3

peler l'histoire de la société SMT. Je présenterai ensuite, dans un second chapitre, les trois G3 et leur contexte d'acquisition, avant de conclure, en troisième partie, sur ce qu'ils représentent dans l'histoire de l'industrie micro-informatique française.

L'histoire de SMT

L'histoire de la société SMT est l'histoire, somme toute banale, d'un constructeur de micro-ordinateurs en France dans les années 1980, mais elle est aussi empreinte d'une certaine originalité qui la démarque de ses concurrents.

Une histoire parmi tant d'autres

L'apparition sur le marché des composants en silicium permettant de concevoir soi-même un micro-ordinateur a incité des entreprises d'électronique françaises à intégrer ce nouveau secteur de l'industrie informatique. Ainsi, R2E, entreprise d'électronique industrielle, a produit le Micral en 1973, une année après l'apparition du micro-processeur 8008 (Petitgirard, 2020). HBN, producteur de composants électroniques à Reims, a construit le Guépard en 1984. MTB, constructeur à façon de produits électroniques, a créé la société de micro-informatique Apollo 7 en 1984. Thomson et Matra se sont essayés

également dans ce nouveau marché. Certains professionnels ou amateurs d'électronique ont tenté eux-aussi personnellement l'expérience en créant leur propre entreprise. Georges Cottin, ancien employé de HP, et Jean-Pierre Bouhot, rédacteur en chef du magazine *L'industrie nouvelle*, ont lancé la société MBC en 1975, en même temps que leur premier micro-ordinateur, l'Alcyane. La société SMT (Société de Micro-informatique et Télécommunications) a vu le jour le 18 décembre 1979. Son président était Claude Madeleine-Perdrillat, et Joseph Rinaudo, son directeur³, deux anciens de la Direction Générale des Télécommunications⁴.

Chaque société a défini l'architecture matérielle de sa machine en fonction du marché qu'elle visait. Pour la société Logystem et son Ecritemap (1982), ce fut le secrétariat ; pour HBN Computer et son Guépard (1984), ce fut le développement⁵. SMT s'est naturellement orienté vers le monde des télécommunications.

Face à la très forte concurrence étrangère, et notamment américaine, le gouvernement français a adopté une poli-

tique industrielle interventionniste vis-à-vis des constructeurs nationaux. Elle s'est, entre autres, matérialisée dans des plans d'éducation à l'informatique s'appuyant sur la production nationale. Ce ne fut pas le seul pays européen à pratiquer ainsi : en 1981, la Grande-Bretagne a choisi Acorn Computer et son BBC Micro, pour son programme d'initiation à l'informatique sur la BBC ; la Suède a introduit le Luxor ABC 80 puis l'ABC 800, dans ses écoles pour son programme d'enseignement de l'informatique. En France, les premiers micro-ordinateurs ont intégré les collèges et les lycées avec l'opération 10000 micros (1978) puis les écoles en 1983. Sont venus ensuite, le plan Informatique pour tous (IPT) en 1985 (Boule, 1984). Comme Thomson, Excelvision, Logabax, Bull-R2E, Matra et Léonard, entreprises nationales ou dont la production est nationale, SMT va profiter du plan IPT.

D'autres formes de subventions ont existé aussi bien pour le matériel que pour le logiciel. MTB a ainsi bénéficié d'une subvention de l'Agence Nationale de Valorisation de la Recherche (ANVAR) pour la construction du Squale, en 1984⁶. SMT a par ailleurs signé une convention avec l'État français pour l'« *adaptation de progiciels horizontaux aux standards français dans l'environnement graphique Windows ou dans l'environnement réseau MS-NET* »⁷.

³ Extrait du registre du Commerce et des Sociétés, Fonds SMT-Goupil, musée des Arts et métiers, Boîte 40853-0002- 2/7, Dossier « Documents divers ».

⁴ « Sur le grill, Claude Perdrillat, SMT Goupil », *Ordinateurs*, 9 septembre 1985, pp 28-38, Fonds SMT-Goupil, musée des Arts et Métiers, Boîte 40853-0006-1/5, dossier G4.
Bernard Werber, « Le Roman de Goupil », *L'Obs Économie*, 12/09/1986, pp 70-71 [URL : http://referentiel.nouvelobs.com/archives_pdf/OBS1140_19860912/OBS1140_19860912_070.pdf].

⁵ *Décision Informatique*, n° 75, 6 mai 1985, p. 87.

⁶ Note de M. Mitjaville à M. Subremon, Objet : « société des Matériels Techniques de Boulogne (M.T.B.) », 6/08/1985, AN 19860179/5, Dossier IPT.

⁷ Convention 85 35 537, 31/12/1985, AN 19910552/2. Un « progiciel horizontal » est un produit logiciel non dédié à une profession ou une application particulière,

Ces efforts n'ont cependant pas permis de maintenir l'industrie micro-informatique nationale. La concurrence des produits américains et japonais, la taille du marché français sans commune mesure avec le marché américain, empêchant de rentabiliser les productions sur le marché national, la difficulté des sociétés françaises à exporter, le changement de politique industrielle du gouvernement français à partir de 1986 ont fait disparaître les entreprises les unes après les autres. La société Apollo 7 qui commercialisait le Squal de MTB, n'a pas survécu au fait qu'il n'ait pas été sélectionné pour le plan Informatique pour Tous. Thomson jettera l'éponge en 1989, deux ans avant SMT.

Une certaine originalité

La société SMT montrait cependant plusieurs originalités.

La première est celle du contexte de sa naissance, raconté par Claude Perdrillat lui-même, dans un article datant de 1980 (Perdrillat, 1980). La société SMT a vu le jour au sein du club Microtel d'Issy-les-Moulineaux pour industrialiser et commercialiser un micro-ordinateur qui y fut conçu. Les clubs Microtel constituaient une fédération de clubs de micro-informatique dont les adhérents étaient des amateurs débutants,

des bricoleurs confirmés ou des semi-professionnels, tous passionnés de micro-informatique. Ces clubs étaient soutenus par la Direction générale des télécommunications (DGT), le Centre national d'étude des télécommunications (CNET), respectivement service et centre de recherche du ministère des PTT. Cette aide se manifestait par le prêt de locaux pour se réunir, mais également par la présence de professionnels, comme ceux des laboratoires du CNET ou de l'université Paris 6 qui, durant l'été 1979, ont apporté leur soutien au club Microtel d'Issy-les-Moulineaux dans leur projet de création d'un micro-ordinateur. Le prototype⁸ fut fabriqué et commercialisé par la société SMT sous le nom de Goupil G1, en 1980, année où Joseph Rinaudo était le président du club d'Issy-les-Moulineaux (*ibid.*, p. 56).

L'intervention d'amateurs et de professionnels dans la création du produit est maintenue lors de la conception des micro-ordinateurs suivants. En 1982, l'année de la sortie du G2, SMT avouait que les adhérents des clubs Microtel constituaient toujours « *un banc d'essai extrêmement sélectif, [et] particip[ai]ent à la promotion et à la création originale qui entoure le produit* »⁹. Cette même année, la société Logsystem communiquait sur sa capacité à développer l'Écritext en étroite coopération avec ses utilisateurs finaux, les secré-

par exemple, un traitement de texte, un tableur ou un gestionnaire de base de données (Sportiche et Defrasne, Note de présentation de la convention 85 35 537, 08/12/1985, AN 19910552/2).

⁸ Présent dans les collections du musée, Inv. 43123-0001-.

⁹ Plaquette de présentation SMT, 1982, Fonds SMT-Goupil, musée des Arts et Métiers, Boîte 40853-0013- 5/5.

taires¹⁰. Mais comme Gilles Garel (Garel, 2018), j'estime que c'est à une tout autre implication que SMT invitait les utilisateurs des clubs Microtel ; en effet, l'aveu de SMT laisse voir un véritable travail de co-conception.

Le soutien du ministère des PTT aux clubs Microtel et le passé professionnel de Perdrillat et Rinaudo, ont défini la politique industrielle de SMT, résolument tournée vers les télécommunications. Ainsi, le G1 (1980) et le G2 (1981) pouvaient être reliés au réseau téléphonique analogique. La communication numérique est apparue en 1983 avec le G3 qui intégrait la norme internationale vidéotex. C'est cette originalité que Perdrillat a défendue en 1985, lorsque la compatibilité avec la plateforme d'IBM s'est avérée incontournable¹¹. Le G3PC (1984) et le G4 (1985), dit compatible avec l'IBM PC AT, ont continué à jouer la carte de la communication. Ils possédaient chacun une carte réseau Omninet, capable de relier des micro-ordinateurs Apple et TRS80, d'architectures matérielles différentes. Pilotée par le logiciel MS/NET, elle constituait le réseau GoupilNet, pouvant connecter jusqu'à 64 machines, d'après les indications du constructeur. Cette capacité à l'hétérogénéité matérielle du réseau permet à SMT de se démarquer d'autres constructeurs de micro-ordinateurs, tels Apple ou Bull-

R2E, qui, s'ils disposaient d'une interface réseau, étaient souvent reliés par un réseau propriétaire. Les machines G40 (1986) puis G50 (1988) étaient quant à elles des serveurs de fichiers, capables de servir les 64 machines d'un réseau GoupilNet.

Cette recherche d'originalité technique n'a pas suffi cependant à sauver la société. Les réseaux vont devenir indispensables et la solution adoptée par SMT, qui pensait qu'elle deviendrait une norme, s'est avérée peu judicieuse. Comme d'autres, elle fut balayée par la combinaison des protocoles TCP et IP, présente sur les systèmes UNIX avant de diffuser vers les autres systèmes d'exploitation.

À partir du G4, SMT a adopté une politique de compatibilité matérielle avec les plateformes successives du micro-ordinateur d'IBM : le G5 (1987) et le G6 (1990) étaient tous deux totalement compatibles AT. La société a maintenu sa politique d'originalité sur un autre créneau, celui des ordinateurs de bureau transportables haut de gamme avec le compatible Club (1986) puis le Golf (1988). Sans succès. La firme disparaît en 1991.

Les trois G3 des collections du musée des Arts et Métiers

Les trois G3 n'ont pas été acquis ensemble. Deux appartiennent au fonds SMT-Goupil, confié au musée en 1991 par le directeur de la communication de SMT, au nom de la société, afin que demeure « *une histoire de l'entre-*

¹⁰ Micro7 n°07, juillet-août 1983, p. 87.

¹¹ « Sur le grill, Claude Perdrillat, SMT Goupil », *Ordinateurs*, 9 septembre 1985, pp. 28-38, Fonds SMT-Goupil, musée des Arts et métiers, Boîte 40853-0006-1/5, dossier G4.

prise »¹². Le troisième est entré en collection avec un nanoréseau du plan Informatique pour tous (IPT) dont il est le serveur. Nous ne connaissons pas l'histoire du nanoréseau au sein de l'école, je me contenterai ici de le décrire avant de présenter plus longuement dans une partie spécifique, le fonds SMT-Goupil.

Le nanoréseau fut proposé par une école élémentaire de Nancy au musée des Arts et Métiers en 2008. Celui-ci l'accepta dans ses collections car d'une part, il était un souvenir du plan IPT et évoquait l'histoire des réseaux locaux en France, mais aussi parce qu'il était absolument complet. Outre son serveur et les logiciels, il comprend non seulement les six postes de type MO5 de Thomson, leur stylo optique, leur cartouche de mémoire RAM, leur bloc d'alimentation, leur boîtier de configuration en poste de nanoréseau et leur boîtier de communication pour les raccorder au nanoréseau, mais également des pièces qui semblent plus insignifiantes malgré leur importance : les deux bouchons de terminaison du réseau, et les câbles réseaux. Le nanoréseau était également équipé de deux imprimantes et de deux lecteurs-enregistreurs de bandes magnétiques.

- *Le fonds SMT-Goupil*

Ce fonds comprend des archives administratives présentant l'évolution de la société à travers ses rapports d'ac-

tivité et ses bilans internes, les études et les documents reflétant les événements clés de la vie de la société (augmentation du capital, inauguration de ses antennes régionales et ses bureaux internationaux), la mise en place des procédures internes.

Il intègre des archives commerciales, conservant les initiatives marketing, dont les campagnes publicitaires du G3 ou les recherches sur le logo de la société, l'évolution de la communication extérieure et interne de la société, ses politiques de vente et de distribution.

Il conserve également des archives techniques regroupant l'ensemble des manuels associés aux micro-ordinateurs produits par la SMT, des documents supports des produits, des photographies des chaînes de montage, des études contrôle-qualité sur les produits. Le fonds iconographique et multimédia est important, tant sur les produits que sur les événements commerciaux jalonnant la vie de la société (*cf. fig. 2*)¹³.

Les documents constituant ce fonds remarquable, qui mériterait d'être analysé en profondeur, ont systématiquement été inventoriés par le centre de documentation du musée¹⁴.

¹² Dépêche AFP, 10 juillet 1991 « Goupil : sauf miracle, l'acte de mort sera prononcé jeudi », Dossier d'œuvre 40841-40853, musée des Arts et Métiers.

¹³ Fonds SMT-Goupil, musée des Arts et Métiers, Boîte 40853-0010- 3/6, Dossier « Planches et photos, signature contrat ».

¹⁴ Les fonds inventoriés sont consultables sur le site du musée des Arts et métiers [URL : <https://www.arts-et-metiers.net/musee/archives>].



Figure 2 - à gauche : Claude Perdrillat et Bill Gates, lors du dîner du Pré-Catelan, 9 mai 1985 ; à droite : au premier plan Bill Gates et Joseph Rinaldo, lors de la signature du contrat. Au second plan, Bernard Vergne, directeur général de Microsoft France et Roger Abehassera de Microsoft

Il inclut également chacune des machines produites par l'entreprise, y compris l'unique exemplaire de la version portable du Goupil 1, présentée au SICOB en 1981. L'ensemble du matériel permet de retracer l'histoire des choix de la société. Ce fut tout d'abord ceux (1) d'une architecture ouverte, puisque dès le G1, l'utilisateur avait la possibilité d'ajouter des cartes électroniques, pour peu qu'elles existent¹⁵, afin d'augmen-

ter les fonctionnalités du micro-ordinateur, et (2) d'une forte orientation vers les télécommunications : le G1 et le G2 possèdent des modems acoustiques, remplacés par des cartes réseaux pour tous les modèles suivants, transformant certains de ces derniers en serveurs multipostes.

À partir du G3 s'est affirmée l'orientation vers une micro-informatique professionnelle haut de gamme avec l'abandon du facteur de forme monobloc des G1 et G2 pour une structure modulaire plus pratique (*cf.* fig. 1), affichant matériellement, par son esthétique, l'influence de l'IBM PC. Cette esthétique

15 P. Hurbain, « Comment passer du 48 au 64K ? », *Goupil Revue*, n° 1, mars 1982, p. 36 [URL : http://download.abandonware.org/magazines/GOUPIL%20Revue/goupilrevue_numero01/Goupil_Revue%201_0036.jpg].

est particulièrement soignée. Pour le G3, elle est l'œuvre de Roger Tallon¹⁶, célèbre designer, auteur, entre autres, du design du train Corail ou des TGV. Pour le micro-ordinateur portable Golf (1990), le travail de Dominique Delamour, de la société Genus fut récompensé par l'Oscar du design en 1989¹⁷. Le G4 puis surtout le G5 reflètent la décision de SMT de perdre sa spécificité matérielle pour adhérer aux standards Intel-IBM-Microsoft, tout en maintenant la particularité de produire des micro-ordinateurs communicants. Mais à bien y regarder, cette tentative de ménager le standard et l'originalité était amorcée déjà avec les G3.

• *Les Goupil G3*

Il faut parler des G3 plutôt que du G3. En effet, les trois exemplaires du G3 appartenant aux collections du musée possèdent chacun une configuration différente.

Le G3 de 1983, est un ordinateur multiplateforme. Il est le pur produit de la politique que SMT se donne en cette année 1983 : résister à la concurrence mais également s'aligner sur les standards ; afficher 35 % de spécifique et 65 % de standard¹⁸. D'architecture

ouverte comme ses prédécesseurs, il dispose de sept cartes insérées dans son châssis dont trois cartes processeurs, l'une supportant un 6809, une autre un Z80 de Zilog et la troisième le 8088 d'Intel, les trois microprocesseurs les plus fréquents du milieu de la décennie 1980. Il était possible de les insérer deux à deux dans le capot du G3. L'intérêt de cette configuration multiplateforme était que l'utilisateur avait à sa disposition deux environnements de travail différents qu'un interrupteur permettait de sélectionner¹⁹. La carte supportant le 6809 de Motorola conservait une compatibilité ascendante avec les modèles précédents de SMT, construits autour d'un processeur de la même famille. Le Z80 assurait la disponibilité du système d'exploitation CP/M et de sa très enviable bibliothèque de logiciels. La carte équipée du 8088 d'Intel permettait la compatibilité logicielle avec l'environnement MS-DOS et le monde d'IBM.

Cette configuration multiplateforme était assez exceptionnelle, même si elle ne fut pas unique. D'autres entreprises ont choisi cette option cette même année, comme DEC avec le Rainbow ou Epson et son QX-10, intégrant deux micro-processeurs, le Z80 et le 8088 dans leurs micro-ordinateurs.

¹⁶ Archives Roger Tallon, Musée des arts décoratifs, ISMGOU8140003 Tallonlot1, ISMGOU8240009-01 à 02 Tallonlot2, SMT-6 Tallonboite52, SMT-7 Tallonboite52.

¹⁷ Fonds SMT-Goupil, musée des Arts et Métiers, Boîte 40853-0002- 1/7, Dossier « Diverses correspondances trouvées dispersées », sous-dossier « oscar du design ».

¹⁸ Fonds SMT-Goupil, musée des Arts et Métiers, carton 40853-0013- 5/5, Dossier 1983, chemise

« Résultats 1983, dossier de présentation presse mars 1984 » p. 3.

¹⁹ « Le Goupil G3 », *L'ordinateur individuel*, n° 49, pp. 166-171 [URL : <http://fr.1001mags.com/parution/l-ordinateur-individuel/numero-49-juin-1983/page-170-171-texte-integral>].

Le G3PC, sorti une année plus tard, ne possède plus qu'une carte processeur, celle supportant le 8088. Elle accompagne six cartes, cinq se trouvant déjà dans la configuration du G3, auxquelles s'ajoute une carte réseau Omninet, permettant de donner au G3PC un rôle de nœud de réseau GoupilNet, voire de serveur.

C'est cette configuration, disponible dès 1984, qui permettra à la société de participer au plan Informatique pour Tous. En effet, les spécifications matérielles du serveur du nanoréseau précisaient que celui-ci devait être un micro-ordinateur professionnel, ouvert sur les standards du marché pour supporter des logiciels différents et capables de dialoguer avec des périphériques. Le matériel devait être éprouvé et déjà en production, afin qu'il soit compétitif et totalement banalisé²⁰. Leur valeur ajoutée devait être réalisée en France²¹ et le constructeur devait disposer d'une capacité de production suffisante pour équiper les établissements avant la rentrée 1985²². Apple, intéressé par le marché, ne fut pas choisi car les délais lui imposaient d'importer sa machine d'une de ses usines à l'étranger, le temps manquant pour en construire une en France. IBM, DEC et HP avaient été refusés lors

de la constitution des marchés publics antérieurs pour des questions techniques de réseau, de prix ou de manque de compatibilité²³. Bull, SMT, Léanord et Matra se sont partagé le marché.

5000 machines spécifiques seront commandées à SMT, pour les nanoréseaux des collèges et des lycées, car le G3PC incluait le langage LSE dans sa configuration de base, diminuant ainsi le coût pour le plan IPT²⁴. C'est pourquoi la configuration du G3PC du nanoréseau (1985) s'avère identique à celle du G3PC à la carte réseau près : la carte Omninet est remplacée par une carte nanoréseau de Léanord.

Conclusions

Loin d'être une redondance, l'existence de trois micro-ordinateurs Goupil G3 est un marqueur de l'histoire de la société SMT mais également de l'histoire de l'industrie micro-informatique en France. Elle témoigne du changement de politique de la société de constructeur informatique : depuis sa création, celle-ci visait le marché des micro-ordinateurs professionnels en entreprise. Avec le G3,

²⁰ Rapport de présentation au comité directeur du FIM, Annexe II : « L'offre en micro-informatique », 13/11/1984, AN 19860179/5, dossier Informatique pour Tous, p. 1.

²¹ Note n° 67-13, de M. Cholley à M. Mitjaville, non datée, AN 19860179/5, dossier Informatique pour Tous.

²² Note de M. Mitjaville à Mme le ministre du MRICR, 10/01/1985, AN 19860179/5, dossier Informatique pour Tous, p. 1.

²³ Note pour le ministre, en provenance du ministère du budget. Objet : « Commandes de micro pour l'enseignement et procédures de marchés publics », 28/02/1987. AN 19860179/5, dossier Informatique pour Tous.

²⁴ Note du Délégué chargé des nouvelles formations, M. Trigano, au premier ministre. Objet : « Chiffrage du matériel IPT ». 15/03/1985. AN 19860179/5, dossier Informatique pour Tous.

la visée est visiblement autre. SMT évolue vers un marché de micro-ordinateurs haut de gamme. Le changement de forme des machines, passant d'un monobloc ni pratique, ni esthétique, à un système hautement modulaire, l'appel à un designer de renom pour réaliser cette évolution : tout montre que l'entreprise souhaite offrir un produit élégant. Les efforts marketing que la société déploie aussi bien vers le monde de l'entreprise que vers les centres de recherches et l'enseignement indiquent son souhait de pénétrer de nouveaux marchés. Enfin, SMT joue pleinement du choix d'architecture ouverte de ses produits, pour s'adapter au marché de la micro-informatique et aux opportunités du marché local. Avec le G3, le constructeur ouvre la configuration de ses micro-ordinateurs en acceptant de gérer, en plus de celle qui faisait sa spécificité, des configurations matérielles ou logicielles standardisées par le marché, puis, avec le G3PC, de se limiter au standard IBM PC, avant de s'adapter de nouveau pour l'importante commande que représentait le plan Informatique pour Tous, avec le G3PC nanoréseau.

Le G3 peut être ainsi considéré comme un point de basculement de la société SMT, mais également des principaux constructeurs français de micro-informatique, vers une architecture standardisée. Comme Léanord, Bull-R2E, MBC, SMT avait initialement construit ses machines autour de choix techniques assumés. À partir de 1984, tous, y compris MBC devenu Axel puis Matra, vont adopter l'architecture proposée par IBM, Intel et Microsoft, devenue un standard de fait.

Bibliographie

Boule F. (1984). « Informatique à l'école, Introduction et éléments d'histoire » [en ligne]. Dossier EPI n° 6, Supplément au *Bulletin de l'EPI*, septembre [URL : http://referentiel.nouvelobs.com/archives_pdf/OBS1140_19860912/OBS1140_19860912_070.pdf].

Garel G. (2018). « Les communautés aux origines de la micro-informatique : des amateurs aux entreprises » [en ligne]. Actes des 9^e Journées du Groupe Thématique Innovation de l'AIMS, « Communautés, écosystèmes et innovation », oct. 2018 Montréal, Canada, p. 11 [URL : <https://hal-cnam.archives-ouvertes.fr/hal-02557212/document>].

Petitgirard L. (2020). « Biographie de l'ordinateur R2E-Micral, ou comment faire exister un 'micro-ordinateur' dans les années 1970 » [en ligne]. *Technologie et innovation*, 5 [URL : https://www.openscience.fr/IMG/pdf/iste_techinn20v5n2_1.pdf].

Perdrillat C. (1980). « L'innovation en matière de micro-processeurs à domicile ». *Culture technique*, 3, 1980, pp. 56-57.

La micro-informatique, outil et enjeu d'innovations dans les formations du Cnam (1970-1990)

Camille Paloque-Bergès

HT2S, Cnam

Loïc Petitgirard

HT2S, Cnam

Introduction

À la fin des Trente Glorieuses, le Conservatoire national des arts et métiers est devenu un établissement qui assume dans son contexte de formation et de recherche l'intégration organique d'un nouveau domaine universitaire, les « sciences pour l'ingénieur » (Ramunni, 2005). C'est aussi un lieu qui renouvelle sa tradition d'interaction avec les professionnels via les nouveaux principes des « sciences de l'action » (comprenant les sciences économiques, de gestion et de l'organisation, de l'information et de la communication, etc.), impliquant la mise en regard des savoirs avec les pratiques de terrain. Les formations à l'informatique côtoient les formations dans d'autres disci-

plines elles-mêmes augmentées par l'équipement technologique informatique dans le secteur des sciences pour l'ingénieur (la mécanique, la métrologie, la chimie ou l'électronique) comme dans la comptabilité, les travaux documentaires, etc., pendant « appliqué » des sciences de l'action.

Les évolutions du positionnement du Cnam en matière d'informatique se fondent sur un contexte économique qui stimule la recherche de nouvelles technologies pour faire face à la crise économique – alors que le Club de Rome signale dans son célèbre rapport de 1972, « Les limites de la croissance »¹. En France, la présidence

¹ Meadows D.H., Randers J. & L. Meadows D.L. (1972). *The limits to growth : A report for the Club of Rome's*

Pompidou (1964-1974) se fonde sur une logique du plan – notamment, pour le volet informatique, dans les pas du Plan Calcul initié par son prédécesseur (Mounier-Kuhn, 2010) –, suivi par le tournant libéral et la politique des créneaux ouverte sur l'Europe de Valéry Giscard d'Estaing (1974-1981). Parce qu'il est régulièrement concerné par ces évolutions de politique des sciences et technologies, économique et industrielle, voire même directement sollicité, le Cnam adapte son offre aux nouvelles demandes.

Dans ce contexte, le tournant « micro » en informatique est négocié au cours des années 1970 de différentes manières par les disciplines, les instituts et les départements qui structurent l'offre de formation au Cnam. Depuis le début de cette décennie, les pionniers de la discipline contribuent à la formation d'une nouvelle génération d'ingénieurs et de chercheurs. La période correspond à la transition de la machine-ordinateur-calculateur à la machine adjuvante de l'organisation de l'entreprise : c'est une transition de l'instrument de calcul à l'instrument de gestion. Par ailleurs, les nouveaux systèmes informatiques convergent vers une approche « répartie » et dialoguent avec les systèmes de télécommunications, au gré du déploiement des réseaux numériques, domaines innovants auxquels le Cnam participe résolument par le biais de petites équipes d'enseignement et de recherche (Paloque-

Bergès & Petitgirard, 2017). Le tournant « micro », la miniaturisation avancée des composants et équipements informatiques qu'il indique, est aussi une manière de reconceptualiser les nouveaux cadres économiques dans une société libérale, dans laquelle on anticipe que l'informatique est en train de jouer un rôle social accru. Les plus enthousiastes n'hésitent pas à proclamer qu'une révolution micro-informatique est en cours.

Parce qu'il est une fenêtre ouverte sur les rapports entre sciences, industrie et société, le Cnam va être un acteur non négligeable dans la conception, la reconnaissance, l'accompagnement, et la diffusion d'une informatique en voie avancée de miniaturisation. Dans cet article de synthèse, issu de recherches préliminaires sur les parcours biographiques de plusieurs professeurs, et sur la formation et l'évolution des départements pédagogiques et laboratoires scientifiques de l'établissement dans le domaine², nous proposons de

2 Outre les fonds d'archives du Cnam concernant le laboratoire d'informatique du Cnam, le Cédric, exploité pour la publication des résultats du projet « Hist.Pat. info.Cnam » (Paloque-Bergès & Petitgirard, 2017), et les dossiers de Professeurs de chaire analysés pour la préparation du prochain *Dictionnaire biographique des Professeurs du Cnam (1955-1975)*, en collaboration avec Claudine Fontanon et André Grelon, nous avons également eu accès aux archives du Cnam au sujet de la métrologie (archives de la chaire de Métrologie, de l'Institut national de métrologie ; archives du Conseil de perfectionnement concernant la mise en place de la chaire d'instrumentation numérique). Au sujet de Bruno Lussato nous avons exploité les archives administratives (dossiers de la chaire de Théories et Systèmes d'organisation) ainsi que les fonds de la Bibliothèque du Cnam où sont conservés plusieurs de ses ouvrages et cours donnés au Cnam.

project on the predicament of mankind. New York : Universe Books. Voir aussi la traduction en français d'une version actualisée du rapport : Meadows D. H., Meadows D. L., Randers J. & Behrens III, W.W. (2012). *Les limites de la croissance*. Montréal : éditions Écosociété.

montrer comment la micro-informatique dans tous ses aspects est saisie au Cnam et par quels acteurs, faisant de cet outil et de ce tournant un enjeu d'innovations dans les formations. Nous nous concentrons sur la période où la miniaturisation de l'informatique prend son essor à la fois dans les domaines scientifiques et dans une nouvelle économie du marché des ordinateurs, dans les années 1970 et 1980. Nous verrons que le passage par le domaine émergent de la micro-informatique, à partir de recherches et d'expérimentations d'enseignants-chercheurs, sort de la salle de cours pour déployer un regard plus large sur les évolutions sociales dans leurs rapports aux organisations et aux politiques économiques plus généralement.

La miniaturisation et l'expérimentation : l'ordinateur, de l'instrument au bureau

La miniaturisation des processeurs des ordinateurs est une constante des années 1960 qui trouve un nouvel élan avec une invention commercialisée en 1971 : le microprocesseur (Lécuyer, 2006). Il est le produit de la jeune firme Intel, au cœur de la Silicon Valley, spécialisée dans la conception de circuits intégrés en semi-conducteur. Intel vante sa nouvelle puce électronique, promet des applications nombreuses, mais si l'intérêt pour le microprocesseur est avéré dès son lancement sur le marché, les utilisations

concrètes mettront un peu de temps à se réaliser : il faut inventer et développer toutes ces applications, et, parmi elles, le micro-ordinateur n'est pas la première. En effet, l'invention et le développement des microprocesseurs sont marqués par une très forte dynamique d'innovation de rupture caractérisés par une série d'« incertitudes cognitive, matérielle et prospective » (Lécuyer & Choi, 2012, p. 48), propices à l'expérimentation.

Au milieu de la décennie, les applications se précisent. En France, la société R2E dépose en 1973 le brevet pour un ordinateur en temps réel, « embarqué »³, qui en plus d'être peu onéreux intègre un système de contrôle : c'est la base du Micral-N commercialisé la même année par R2E (Petitgirard, 2020). Si le brevet annonce les micro-ordinateurs, la technologie du microprocesseur permet d'abord de miniaturiser les systèmes de contrôle de processus industriels, mais aussi des télétransmissions (notamment des données télévisuelles), ainsi que l'instrumentation pour le calcul scientifique. Elle relève d'une informatique « cachée », au sens où le circuit programmable miniature est intégrable : le microprocesseur

³ Le système conçu par R2E est initialement une réponse à une commande passée par l'INRA (Institut national de recherche agronomique) pour un dispositif de mesure automatisée utilisable en plein champ (pour des mesures sur des cultures agricoles). Le Micral est d'abord un ordinateur complètement intégré dans l'instrument, avant d'être commercialisé séparément : mais sa conception (sans écran, sans clavier) en fait un ordinateur destiné à être embarqué dans un système – qu'il soit un instrument scientifique, ou un dispositif de contrôle de processus industriel.

est un « ordinateur sur une puce », selon le slogan choisi par Intel. Les utilisateurs finaux ne perçoivent plus le traitement informatique permettant le pilotage des processus industriels comme des systèmes de leur quotidien (automobiles, ascenseurs). Cela change également la conception des objets et des instruments techniques en renforçant l'importance de la programmation des puces et des logiciels associés : les microprocesseurs sont des dispositifs miniatures réalisant très vite des opérations simples, et dont la programmation permet la souplesse et la grande diversité d'applications.

Ce cadre technologique apporte son lot de promesses de transformations et appelle à un renouvellement des formations, dont le Cnam est un acteur depuis les années 1950. Il est aussi un cadre très ouvert et propice aux expérimentations, dans les champs des sciences de l'ingénieur comme dans les sciences de l'action.

Le Cnam : une offre de formation étendue

En 1970, les formations en informatique au Cnam couvrent un spectre très étendu allant du domaine des mathématiques appliquées et de la programmation, à l'électronique (technologie des ordinateurs), à l'informatique de gestion, la comptabilité ou encore la théorie de l'information (systémique). Elles trouvent leurs racines dans les années 1950 lorsque se développent des formations à la mécanique, au calcul numérique et l'au-

tomatique, à l'interface avec le monde économique et industriel (Paloque-Bergès & Petitgirard, 2017).

La chaire de Mathématiques en vue des applications, détenue par Alexis Hocquenghem depuis 1950, est liée au département de Mathématiques, principal opérateur des formations à l'informatique. Les années 1960 voient une demande accrue en besoins de formation dans ce domaine, et atteint jusqu'à 14 enseignements, dont les « Machines mathématiques » (Paul Namian), la « Recherche opérationnelle » (Robert Faure), l'« Analyse numérique » (Raymond Theodor). L'évolution de ces formations aboutira à la redéfinition du département de Mathématiques-Informatique en 1967.

La discipline de l'informatique se dessine progressivement à l'échelle nationale, non sans de vifs débats institutionnels et épistémologiques, et au niveau local notamment sous l'influence de la chaire d'Informatique-programmation détenue par François-Henri Raymond depuis 1973. Ce dernier est fondateur de la Société d'Électronique appliquée à l'Automatisme (SEA), pionnière en France dans la construction industrielle d'ordinateurs. Il jouit d'une notoriété d'expert bien au-delà de l'établissement, et avait déjà été un temps au Cnam, recruté en 1946 comme chef des travaux par Albert Métral, titulaire de la chaire de Mécanique. Ensemble, ils participent à l'essor du domaine de l'Automatique (qui s'oppose à la Cybernétique) en France (Le Roux, 2018). En 1956 se tient d'ailleurs au

Cnam le Congrès international d'Automatique, dont Raymond est un organisateur.

Suite au lancement du Plan Calcul en 1966, la SEA est absorbée en 1969 dans le champion national en construction, la CII (Compagnie Internationale d'Informatique) à l'encontre des projets formulés par F.H Raymond. Celui-ci revient au Cnam en 1973, sur une chaire taillée à sa mesure et accueilli au département de Mathématiques-Informatique par son collègue et ami Hocquenghem.

Une autre demande devient pressante au fil des années 1960, celle émanant du secteur tertiaire, non plus seulement au niveau des calculs de comptabilité, mais à toutes les échelles de la gestion des entreprises. L'Institut d'Informatique d'Entreprise (IIE) est créé en 1968 au sein du Cnam, précisément pour délivrer une formation à l'informatique de gestion, et pour répondre aux besoins du nouveau marché de l'informatique qui a fortement évolué depuis les métiers de la mécanique (Neumann, 2013).

Enfin les formations en électronique des impulsions au sein du département Électronique vont évoluer vers la « technologie des ordinateurs ». La formation des techniciens et ingénieurs dans ce domaine s'est faite initialement en association directe avec le constructeur Bull, qui installe en 1964 dans l'établissement un « Centre Cnam-Bull ». Les cycles de formation seront ouverts à d'autres nombreux acteurs (le CEA – Commissariat à l'Énergie Atomique, ONERA – Office

national d'études et de recherches aérospatiales, IBM France, Philips...). Si le département de Mathématiques-Informatique est consulté, c'est initialement au sein des chaires et départements d'Électronique (sous la houlette de George Boutry, Yves Angel puis Michel-Yves Bernard) que les cours se déroulent, selon une ligne directrice fixée par les deux principaux organisateurs et intervenants chargés de cours : Jean-Paul Vabre (Ingénieur de la Division Études de Bull-GE) et Georges Metzger (Chef du Service de Formation technique chez Bull-GE).

Expérimenter sur les machines

Dans ces filières, les machines sont des supports de formation aussi bien que d'expérimentations, voire des deux en interaction. En 1968, le laboratoire de calcul, qui sert les besoins de l'établissement en informatique mais est géré par le département de Mathématiques (puis Mathématiques-Informatique), se dote d'un IBM 360/30, dont l'une des principales utilisations concerne les besoins en calcul du département. Au début des années 1970, le Centre de Recherche et d'Expérimentation pour l'Enseignement des Mathématiques (CREEM) est créé à l'initiative de René Chenon, titulaire de la chaire de Mathématiques appliquées aux arts et métiers. L'IBM 360/30 est la machine support de ces expérimentations ; c'est un ordinateur qui annonce l'ère des mini-ordinateurs, en rupture avec les macro-systèmes du traitement par lot, en même temps que les évolutions

rapides vers une informatique à plus petite échelle, dédiée à multiplier et paralléliser les utilisations et les tâches exécutées (Haigh, *in* Paloque-Bergès et Petitgirard, 2017). Le Centre utilise le langage CourseWriter, développé par IBM à destination de problématiques d'enseignement. Ces expérimentations s'inscrivent dans le cadre de « l'enseignement programmé » visant à perfectionner et automatiser des processus d'apprentissage. Le CREEM porte le développement d'une nouvelle machine à enseigner, le dispositif baptisé MITSI-2003 (Monitrice d'instruction technique et scientifique individuelle), s'appuyant sur son expertise en matière de mathématiques, d'informatique et de didactique de ces disciplines. Le projet expérimental est aussi contraint par une économie de moyens, qui rencontre le besoin de se rapprocher des utilisateurs, en réduisant la taille des machines, ou à défaut, de l'interface entre l'ordinateur et l'utilisateur. Pour réaliser des expériences sur le télétraitement (transfert de données localement et à distance) et les systèmes répartis (le système Unix en priorité), expérimentées dès 1975, des mini-ordinateurs sont acquis, commençant par la série PDP de l'équipementier états-unien Digital Equipment Corporation. Systématisées sous la forme de réseaux hétérogènes à la fin de la décennie, ces activités migreront sur les petites stations informatiques de travail de Sun Microsystems dans la décennie suivante, suivant le mouvement de migration des équipements (notamment pour les réseaux) vers la micro-informatique (Paloque-Bergès & Petitgirard, 2017).

Les technologies miniaturisées, sous forme du composant microprocesseur ou de systèmes informatiques prêts à l'emploi, intéressent d'autres domaines, en particulier celui de l'instrumentation scientifique et la métrologie, sujets portés par l'institut national de métrologie (INM) au Cnam pour la recherche et au sein du Département de Physique-Métrologie pour les formations. À la fin des années 1970, l'industrie des instruments scientifiques sollicite le Cnam pour la création de formations spécifiques : elles seront organisées autour d'une nouvelle chaire d'Instrumentation scientifique à partir de 1982. L'instrumentation est tout à la fois tributaire de l'évolution du système technique et moteur de sa transformation. Au tournant de 1980, l'instrumentation s'annonce déterminante dans la poursuite de l'automatisation avec la robotique qui conjugue des traitements informatiques lourds avec la multiplication des capteurs. La disponibilité du microprocesseur pour les systèmes de traitement de l'information et l'apparition de nouvelles générations de capteurs constituent le cadre technologique de ces transformations.

Le cours général sur les « Instruments de mesure modernes », ou le cours spécialisé de « Conception de systèmes » donnent le cadre général. Les formations plus spécifiques s'attachent à la conception des « Instruments numériques », qui promeuvent les concepteurs d'instruments en intermédiaires entre les informaticiens, les électroniciens et les

physiciens. Le champ est bien interdisciplinaire, car il s'agit de comprendre aussi bien les exigences du traitement du signal que les spécificités des microprocesseurs (de leur programmation à leur utilisation dans le contexte des systèmes de mesure numérique), la physique des capteurs, etc. Point non négligeable, le titulaire de cette nouvelle chaire, Claude Morillon, n'a pas été facile à recruter parmi un très faible nombre de candidats au poste. Son parcours et son action sont à l'image de la convergence tant recherchée. Chercheur au Laboratoire Aimé Cotton (CNRS), il a travaillé sur des systèmes d'acquisition et traitement de données numériques pour la spectroscopie. Ce laboratoire a une forte activité technologique depuis ses origines dans les années 1930, et compte parmi les lieux qui ont su développer leurs propres systèmes de calcul pour perfectionner les instruments scientifiques. Le Cnam confirme par ailleurs sur cet exemple sa position d'interface entre les besoins du monde scientifique, du monde industriel et la construction d'une offre de formations novatrices en France.

Parallèlement, les technologies « micro » sont testées et adaptées à des besoins d'utilisateurs moins scientifiques ou relevant d'industries spécialisées. Dans la lignée des premiers micro-ordinateurs français Micral, le microprocesseur qui va devenir le support novateur des futurs ordinateurs de bureau est l'états-unien Intel 8008. La machine Alvan qui est essentiellement présentée sur le marché français en

1974 est moins générique que le Micral, et plus orientée sur la gestion et la bureautique. Elle a été mise au point par Alain Grébert et développée avec son frère Yvan Grébert (Alvan est la contraction d'« Alain-Yvan »). Elle est munie d'un clavier, d'un écran, d'un système de stockage – assemblage relativement inédit à l'époque – et surtout de son propre langage : la facilité de la communication avec la machine en fait sa singularité et devient l'argument principal de sa commercialisation. C'est aussi une des raisons pour en faire le terrain des expériences du professeur du Cnam Bruno Lussato, qui publie avec Bruno France-Lanord et Jean-Pierre Bouhot toujours en 1974, un ouvrage fondateur de la micro-informatique, *La micro-informatique : introduction aux systèmes répartis*. Cet ouvrage fait suite à la publication d'une plaquette de 60 pages en décembre 1973 consignant les premières remarques de Lussato sur le sujet : « L'informatique à l'envers : les mirages de l'informatique ». En avril 1974, Lussato présente lors de son séminaire au Cnam les machines Alvan et d'autres ordinateurs fabriqués par Hewlett-Packard en démonstration devant des journalistes et chefs d'entreprises. Le micro-ordinateur, selon cette nouvelle conception, ferait ainsi passer du calcul à la bureautique générale. La promotion de la micro-informatique est ainsi portée par les sciences de l'action telles qu'elles se pratiquent au Cnam, en même temps que se développe la réflexion sur les rôles sociaux et économiques de ces technologies miniaturisées.

L'ouverture sur les rôles sociaux et économiques de la micro-informatique

La micro-informatique vient répondre aussi bien aux scénarios de la « société post-industrielle » qu'aux débats sur la « société de l'information ». Ces interrogations se formulent dans le d'un constat de ralentissement de la croissance, et donc, pour les professionnels de l'organisation, au cours d'une réflexion pour renouveler les façons de faire de l'« entreprise du futur » en s'appuyant sur de nouveaux ressorts de croissance et d'innovation – et sur des technologies micro plutôt que macro. En effet, le recours à l'échelle technique du « micro » s'insère dans les nouveaux discours sur les technologies dites « douces », inspirées en France notamment de l'ouvrage *Convivialité* d'Ivan Illich paru en 1973, ou encore de *Small is beautiful*, d'Ernst F. Schumacher, paru la même année (trad. fr., 1979), et qui promet une « société à la mesure de l'homme ». Dans cette seconde partie, nous nous pencherons ainsi de plus près sur les propositions de Bruno Lussato, qui fait de la micro-informatique un sujet majeur de ses recherches et enseignement en sciences de gestion, mais aussi dans une réflexion plus générale sur les politiques économiques outillées par les nouvelles technologies. À cet égard, et particulièrement sur la question précise des activités d'enseignement de la chaire de Théorie et systèmes des organisations détenue par Lussato dans les trois dernières décennies du xx^e siècle, nous renvoyons notre lecteur à l'entretien avec

son assistant Bruno France-Lanord, qui complète cet article de synthèse.

Renouveau de l'OST au Cnam

Lussato est professeur au Cnam de 1970 à 2001. Il se place dans la lignée des chaires d'OST (Organisation scientifique du travail), en parallèle de Jean Gerbier (titulaire d'une chaire d'OST en 1970). Ses cours du Cnam servent de support à deux ouvrages de type manuels parus chez Dunod en 1972 : *Modèles cybernétiques, homme et entreprises* et *Introduction critique aux théories d'organisation*, ce dernier restant un manuel de référence en France pendant plus de deux décennies. « *Il y présente l'informatique selon les structures géantes très intégrées (Integrated Information Management Systems) qu'il étudiait dans les grandes sociétés où il était consultant : Shell-France, Philips* »⁴ : de fait, les théories de l'organisation mises en avant par Lussato dans cet ouvrage restent relativement conventionnelles, considérant encore l'informatique en tant qu'outil de structuration industrielle.

Consultant pour de grandes entreprises (IBM, BHV, Shell, Philips, Gloria...), son savoir-faire est largement reconnu dans l'industrie ; mais il produit aussi une réflexion théorique et empirique dans le champ académique, par exemple

⁴ Luc Marco, in notice biographique de Bruno Lussato pour le *Dictionnaire biographique des professeurs du Cnam (1955-1975)*, à paraître aux éditions Garnier.

en 1968 au Laboratoire de Gestion automatisée (Université de Grenoble) – avec un financement d’IBM. Ses travaux de recherche se font en interaction avec son enseignement technique sur la théorie des systèmes et la théorie des organisations. Il promeut l’utilisation de techniques avancées (électronique, informatique) pour renouveler l’OST ; mais cette idée se place dans la conception plus générale de l’entreprise comme système qui met en relation des hommes, des machines, des moyens. Au-delà de l’application des techniques avancées d’organisation, il invente des méthodes, des outils et des instruments – par exemple le « Tracteur Électronique Téléprogrammé » au service commercial du Bazar de l’Hôtel de Ville (BHV) à Paris.

Il reste huit ans chef de travaux au Cnam comme suppléant de Raymond Boisdé, puis est élu sur une nouvelle chaire, « Théories et Systèmes d’organisation », issue d’un dédoublement de celle de ce dernier, avec le soutien d’André Brunet, directeur à l’Institut national des techniques économiques et comptables du Cnam. Lors de son audition à la chaire d’OST en 1978, le jury remarquera :

En ce qui concerne ses références pédagogiques, il faut mentionner, pour s’en tenir aux cours du Conservatoire proprement dits, que, dans le cadre de la Chaire de M. Boisdé, il a créé et assure depuis plusieurs années un nouvel enseignement de spécialité en matière d’organisation commerciale et qu’il donne à titre bénévole une série de conférences sur l’organisation administrative. – D’autre part, M. Lussato se distingue tant par

ses publications que par sa compétence, compétence marquée par des réussites industrielles notamment dans l’application de techniques avancées d’organisation, qui sont directement issues de ses propres recherches⁵.

Lussato défend l’idée d’une OST renouvelée par la systémique et la cybernétique. L’informatique devient pour lui l’occasion d’une contestation du rôle des « ingénieurs-organiseurs ». Selon Lussato, le champ de l’OST n’aurait pas bien saisi le tournant de l’ordinateur, d’où le développement de l’informatique de gestion dans laquelle l’organisation devient secondaire par rapport à l’informatique. Il propose alors de travailler sur la figure des « organisateurs-informaticiens », matérialisation de sa pensée systémique qui se départit des rôles étanches du technocrate d’un côté, du constructeur d’ordinateur de l’autre ou encore de l’informaticien. À la faveur de ces idées, Lussato se fait le défenseur, à partir de 1974, de la micro-informatique, expression qu’il forge lui-même. Lors d’une première phase (1974-1978), cette promotion se fait à destination des managers, ingénieurs, informaticiens, administrateurs, etc., dans le cadre de formations professionnelles. Dans une deuxième phase, après le rapport Nora-Minc de 1978 « L’informatisation de la société »⁶, il soumet ses idées au débat public, et promeut ses théories auprès de publics élargis.

⁵ Cité in Marco, *op. cit.*, à paraître.

⁶ Rapport en ligne [URL : <https://www.vie-publique.fr/rapport/34772-linformatisation-de-la-societe>].

De la pédagogie aux politiques de l'informatique décentralisée

Lussato est donc un animateur et un tenant de la « révolution » de la micro-informatique, prenant comme point d'appui le Cnam. Le recours à la micro-informatique est le support d'une pédagogie de l'informatique décentralisée, essentiellement tournée – du moins au début – vers un public de professionnels. Pour convaincre les managers, elle est présentée comme un outil de gestion, de décision et de contrôle, inscrite dans un paradigme « techno-scientiste » libéral, et ouverte sur des vues prospectivistes – trois aspects cruciaux de la personnalité publique de Lussato.

Pour Lussato, la micro-informatique viendrait contrer l'un des risques majeurs des grands systèmes informatiques accaparés par les États et les grandes organisations multinationales : la tendance à un centralisme informatique corrélât de possibles dérives technocrates, voire autoritaristes. C'est un débat de l'époque que de prévenir contre les dangers de la « méga-informatique », comme la qualifie cet autre collègue de Lussato au Cnam, Jean-Jacques Salomon, chaire de Technologies et Société en 1978, et avec qui il dialogue (Paloque-Bergès & Petitgirard, 2021). En ceci, Lussato participe à la contestation non pas seulement technique mais aussi sociale, économique et politique du paradigme des macro-systèmes, reposant sur de « gros » systèmes informatiques centralisant l'information, organisant des réseaux de transaction et transmission de

données de manière centralisée, contrôlée et fermée. Les « petits » systèmes présentent au contraire une facilité d'emploi, permettant une gestion décentralisée grâce aux structures de l'informatique répartie. Il qualifie ceci de « taylorisme à l'envers »⁷ – un plaidoyer très critiqué.

Sa vision est soutenue et relayée par des réseaux d'entreprise et par l'association Micro-Informatique (AMI) fondée en 1974 avec plusieurs membres industriels tels que L'Oréal, EDF, Philips, Paribas, CEGOS (ex- Commission Générale de l'Organisation Scientifique du Travail). L'AMI trouve avec le Cnam un appui et une tribune, donnant de l'ampleur au séminaire qui s'est déroulé l'année précédente dans l'établissement et dans lesquels Lussato a pu faire ses démonstrations à partir des nouvelles machines Alvan, Hewlett-Packard⁸, ou encore Wang 2200.

Au sein du Cnam, il s'appuie sur le libéral Laurent Citti, directeur général du Cnam (1974-1977), qui imprime à l'établissement une dynamique du même ordre. En effet, en 1974, les esprits convergent : le jeune Citti suit la politique giscardienne du tournant politique libéral ; les travaux novateurs en organisation sont soutenus par François Dalle (P.-D.G. de L'Oréal). Le département

7 « Le taylorisme à l'envers », Paris, Institut de l'Entreprise, 1977 - (26 pages de bande dessinée et 5 pages de texte, coécrit avec François Dalle et Jean Bouinine).

8 Cf. l'entretien avec B. France-Lanord dans ce même volume.

Économie et Gestion, dominant au Cnam depuis les travaux de Jean Fourastié, se renforce encore, parallèlement à l'émergence des « études de sciences et techniques » sous l'impulsion de Jean-Jacques Salomon (Dray et Hadna, 2021) : celui-ci mène des réflexions prospectives sur les politiques de sciences et technologie, depuis l'OCDE pour laquelle il travaille avant de devenir professeur du Cnam.

L'informatique est en train de devenir un sujet politique, conforté par le rapprochement et les intérêts croisés de l'État et de l'industrie à la faveur du Plan calcul, et plus généralement d'une nouvelle forme de programmation des politiques scientifiques, techniques et industrielles. Mais contre la mise en discours étatique, aboutissant en 1978 au rapport de Nora-Minc, Lussato critique radicalement l'idée d'une « informatisation de la société » qu'il assimile au projet moderne d'une quantification omnipotente (l'ordinateur-Golem) conduisant à l'uniformisation généralisée des sociétés. Cette idée se trouve corrélée (même « modulée » sous Valéry Giscard d'Estaing) au centralisme, étatique, planificateur des 1960-1970, évoqué plus haut.

La miniaturisation (sous la forme principale du microprocesseur, comme sous ses formes dérivées à travers les micro-ordinateurs), est une opportunité pour défendre une alternative, ce qu'il appelle la « *privatique* », comme « *antidote à la télématique* », concrétisation du modèle étatique et monopolistique. Ce modèle de la télématique devient opérationnel dès la

fin des années 1970 sur le réseau TRANSPAC (Schafer & Thierry, 2012). Selon Lussato, la « *privatique* », inspirée d'Ivan Illich (1973), repose sur l'autonomisation de l'utilisateur, la « *privatisation* ». La petite échelle ouvre la possibilité de pratiques d'« *artisanat* » technologique, et de formes de décentralisation dans l'organisation industrielle. Elle soutient la promesse d'innovation et de création (par l'appropriation). Des entreprises semi-nationales comme la Compagnie internationale pour l'informatique (CII, devenant Bull en 1982⁹), privilégiée voire imposée par l'État français pour son équipement public, ne devrait pas copier IBM mais suivre d'autres logiques de conception et de marché – celle de l'ordinateur domestique. On peut noter que Lussato aura peu pensé le paradoxe du Minitel, micro-machine tout entière dédiée à l'utilisateur (mais ce dernier conçu comme consommateur d'information) dont les réseaux prennent pourtant les formes du centralisme étatique et monopolistique tant critiqué. En effet, ses mots sur la vacuité des connexions Minitel (dans plusieurs ouvrages, dont 1989) participent davantage d'une critique sur le manque de réflexion des politiques françaises sur la finalité des programmes technologiques (et son incapacité à être attractif sur le marché mondial) qu'il cherche à imposer plutôt que sur la micro-machine à proprement parler.

⁹ Absorbée par Honeywell-Bull en 1975, elle devient partie de CII Honeywell-Bull, rebaptisé Bull en 1982 (source : Wikipédia francophone [URL : https://fr.wikipedia.org/wiki/Compagnie_internationale_pour_l%27informatique]).

Enfin, dans *Le défi informatique* (1981), Lussato s'essaie à la vulgarisation de ses réflexions sur le rapport de l'informatique à l'entreprise et la société. Il a été sollicité par Jean-Jacques Servan-Schreiber, auteur du *Défi Mondial* (1980), et créateur en 1981 du Centre mondial de l'informatique. Dans *Le défi informatique*, Lussato reprend ses métaphores travaillées dans les années 1970 pour marquer son opposition à l'hégémonie d'IBM d'une part et aux projets de télématique d'autre part. Il simplifie, voire caricature le propos : le « grand chaudron » désignant l'informatique lourde, centralisée associée à la télématique est placée en opposition à ce qu'il prône, le « petit chaudron », qui mise sur la micro-informatique, la dissémination d'ordinateurs individuels, la « privatique ». Ce « petit chaudron » est le seul moyen à ses yeux de continuer le développement d'une société moderne et humaine (dans laquelle l'homme est la mesure de toute chose, et non la machine). Les accointances semblent évidentes avec le Centre mondial de l'informatique, mais celui-ci prend une dimension rapidement politique, un projet vitrine du début du mandat de François Mitterrand. Le Centre embauche Nicholas Negroponte, professeur au MIT. Mais la structure est trop peu efficace, le Centre ne tient pas les promesses attendues par Lussato et surtout ne freine en rien les développements techno-politiques de la télématique pilotée par la Direction générale des télécommunications. Lussato prend ses distances avec le Centre, et il en revient à sa posture d'analyste plus distancié, im-

planté au Clame, continuant à publier ses pamphlets critiques mais sans s'empêtrer dans les projets techno-politiques des années 1980.

Pour conclure

La décennie 1970 a été marquée par l'expérimentation technologique avec la micro, autant en sciences de l'ingénieur que dans les sciences de l'organisation au Cnam – une tendance qu'on retrouve dans bien d'autres lieux, sur les campus états-uniens pour ne citer que les plus connus. Au Cnam, ces expérimentations ont pris volontiers un caractère pédagogique, soit dans une perspective très didactique soit en se définissant comme un effort de diffusion de modes d'usage de l'informatique innovants et comme un accompagnement des réflexions sur les rôles multiples pris par une informatique en pleine, et rapide, évolution.

En schématisant le propos, on peut dire que les années 1980 sont celles de la généralisation et de la démocratisation de l'accès aux micro-ordinateurs, portée par la vague de l'IBM PC, accompagnée ou entrant en résonance avec les projets gouvernementaux en France, sous différentes formes : Centre mondial de l'informatique, programme « Informatique pour tous », généralisation de la télématique et du Minitel. Les réflexions conduites au sein du Cnam, avec leur singularité, ont trouvé un écho dans les années 1970 mais semblent se diluer après 1980. La poussée

technologique liée à la miniaturisation comme le caractère massif des projets gouvernementaux ont quelque peu mis en veilleuse les expérimentations singulières, mais de plus en plus marginales, conduites au Cnam, submergées par des projets industriels et politiques d'une tout autre ampleur.

Bibliographie

Dray V. & Hadna S. (2021). Dossier « Actualité de Jean-Jacques Salomon ». *Cahiers d'histoire du Cnam*, 14/1.

Illich I. (1973). *La convivialité*. Paris : Éditions du Seuil.

Le Roux R. (2018). *Une histoire de la cybernétique en France (1948-1975)*. Paris : Classiques Garnier.

Lécuyer C. (2006). *Making Silicon Valley : Innovation and the growth of high tech, 1930-1970*. Cambridge, Mass. : MIT Press.

Lécuyer C. & Choi H. (2012). « Les secrets de la Silicon Valley ou les entreprises américaines de microélectronique face à l'incertitude technique ». *Revue d'histoire moderne contemporaine*, 59/3, pp. 48-69.

Lussato B., Bouhot J.P. & France-Lanord B. (1974). *La micro-informatique : introduction aux systèmes répartis*. Paris : Éditions d'informatique.

Lussato B. (1989). *Dirigeants, le défi culturel*. Paris : Nathan.

Lussato B. (1972). *Modèles cybernétiques, homme et entreprises*. Paris : Dunod.

Lussato B. (1972). *Introduction critique aux théories d'organisation*. Paris : Dunod.

Mounier-Kuhn P.-É. (2010). *L'informatique en France de la seconde guerre mondiale au Plan Calcul. L'émergence d'une science*. Paris : Presses de l'Université Paris-Sorbonne.

Neumann C. (2013). « De la mécanique à l'informatique : les relations entre catégorisation des techniques, groupes professionnels et transformations des savoirs managériaux ». Thèse de doctorat en histoire de

l'Université, Paris 10-Nanterre.

Paloque-Bergès C. & Petitgirard L. (2017). « La recherche sur les systèmes : des pivots dans l'histoire de l'informatique », dossier des *Cahiers d'histoire du Cnam*, 7-8, n° 12.

Paloque-Bergès C. & Petitgirard L. (2021). « Jean-Jacques Salomon, critique précoce de l'informatisation de la société », *Cahiers d'histoire du Cnam*, 14, n°1, pp. 133-162.

Petitgirard L. (2020). « Biographie de l'ordinateur R2E-Micral, ou comment faire exister un « micro-ordinateur » dans les années 1970s » [en ligne]. *Technologie et innovation*, 2020, vol. 5, n° 2 [URL : <http://www.openscience.fr/Biographie-de-l-ordinateur-R2E-Micral-ou-comment-faire-exister-un-micro>].

Ramunni G. (2005). « La création d'un département d'ingénierie au CNRS » [en ligne]. *La revue pour l'histoire du CNRS*, 13 [URL : <https://journals.openedition.org/histoire-cnrs/1693>].

Schafer V. & Thierry B.G. (2012). *Le Minitel. L'enfance numérique de la France*. Nuvis.

Schumacher E.F. (1979). *Small is beautiful : une société à la mesure de l'homme*. Paris : Éditions du Seuil.

Le Cnam, berceau de la micro-informatique

Entretien avec Bruno France-Lanord

Bruno France-Lanord

Maître de Conférences du Cnam à la retraite

Loïc Petitgirard

HT2S, Cnam

Je souhaite avant ce témoignage rendre hommage à quatre acteurs grâce auxquels j'ai eu la chance de participer à l'aventure de la micro-informatique : Jean-Pierre Bouhot, François Dalle, Bruno Lussato et André Truong Trong Thi.

Parcours au Cnam de 1969 au milieu des années 1980

Après trois années passées en classes préparatoires à l'École Navale, un renoncement *in extremis* pendant le concours à poursuivre dans cette voie, et grâce au hasard d'une rencontre avec

tence de cette nouvelle école, j'ai passé le premier concours d'entrée à l'Institut d'Informatique d'Entreprise et intégré cette école en octobre 1969. Créée au Cnam en 1968, l'IIE était la première école d'ingénieurs définis comme des « informaticiens d'entreprise » en France. Une promotion de vingt-cinq élèves, un corps professoral remarquable et un bon équilibre entre les cours d'informatique, de mathématiques, d'économie, d'organisation, de droit... en faisaient une école qui, outre un excellent niveau technique, offrait une ouverture considérable aux élèves. À la fin des études on savait « tout » de l'ordinateur ou presque, puisque l'on apprenait aussi bien l'algorithmique, le fonctionnement des machines, le système d'exploitation,

plusieurs langages de programmation : Assembleur¹, Fortran, Cobol. Au niveau matériel (*hardware*), j'avais même conçu et réalisé un petit système d'affichage en troisième année.

Pour ne citer que quelques professeurs éminents, j'ai le souvenir de Paul Namian en informatique générale, de Robert Faure en recherche opérationnelle et Bernard Lemaire, son assistant, de René Moreau (directeur du centre de recherches d'IBM France) en théorie des langages, de René Chenon en topologie, de Bruno Lussato en organisation, de Georges Depallens en finance, d'Alain Cabannes pour les bases de données, de Claude Delobel pour la programmation en Cobol et bien sûr de notre directeur des études Étienne Pichat. Je me souviens aussi de Gérard Florin à l'IIE et au centre de calcul doté d'un IBM 360/30². Le cours d'organisation de Bruno Lussato dispensé en première année m'avait considérablement impressionné. Il apportait nombre de nouvelles théories et touchait aussi bien à la théorie des systèmes, à la théorie de l'information, à la psychologie génétique, à la linguistique et à la sémantique générale qu'aux théories d'organisation et aux concepts associés. Il venait d'achever une théorie de l'information psycholo-

gique et allait publier en 1972 le premier ouvrage français de théorie des organisations : *Introduction critique aux théories d'organisation*.

À la fin de la première année d'école, ayant été arrêté pendant un an par une tuberculose, ce temps de repos me permit de lire la quasi-totalité de la vaste bibliographie de ce cours. Tout naturellement, lorsque vint le choix d'un sujet de mémoire de recherche pour la troisième année en octobre 1972, je me tournai vers Bruno Lussato qui me proposa un travail sur les chambres de contrôle, lieux de prise de décisions collectives munis d'outils de présentation d'informations, réalisées par Stafford Beer³. Le travail consistait à concevoir, à partir de ses travaux théoriques exposés dans *Decision and control* (1966), un module d'affichage en couleur analogique et numérique d'informations de contrôle de gestion, constituant la base de tableaux de bord de gestion que l'on pouvait construire par assemblage de modules. Cet appareil pouvait aussi faire de la simulation. La phase de réalisation du prototype d'un module se fit en collaboration avec un ingénieur du Cnam qui disposait d'un ordinateur analogique et qui me donna les rudiments de *hardware* pour mener à bien cette réalisation.

1 L'Assembleur est le langage de programmation le plus proche du langage machine binaire lisible par un humain.

2 La machine 360 fut le premier ordinateur de 3^e génération (utilisant des circuits intégrés) d'IBM, commercialisé en 1965. Le Cnam disposait d'un modèle 360/30 du bas de la gamme s'étendant du modèle 20 au modèle 85.

3 S. Beer (1926-2002) fut professeur à la Manchester Business School et consultant, spécialisé dans la recherche opérationnelle, la cybernétique et les sciences de gestion. Il a développé des théories et des modèles systémiques et cybernétiques pour la gestion des organisations à partir des idées de N. Wiener, W. McCulloch et W. R. Ashby.

Bruno Lussato avait déjà installé une chambre de contrôle au Cnam en 1972 avec le soutien financier des entreprises l'Oréal et Primagaz, ainsi qu'avec la collaboration de deux élèves de l'IIE, Pierre Delpuech et François de Thomas de Labarthe, dans son bureau au 2 rue Conté ; un tableau permettait d'afficher de l'information de haut niveau de synthèse pour dirigeants au moyen de carrousels à diapositives Kodak et grâce à des fonctions d'affichage sophistiquées, en particulier le fondu-enchaîné. Il ne reste de cette chambre que les luxueuses boiserie qui habillent la salle 37.1.43., située au 2 rue Conté⁴.

En mars 1973, Jean-Pierre Bouhot, professeur d'informatique à l'École Centrale, fondateur et éditeur de la revue *L'Informatique*⁵ nous envoie un article publié en décembre 1972 qui l'avait frappé : « A view of computer architecture » de Caxton C. Foster, spécialiste de l'architecture des ordinateurs à l'Université du Massachusetts. Dans son article ce

dernier prévoyait le développement pour les vingt-cinq années à venir de micro-processeurs standardisés et peu coûteux (moins de 100 dollars en 1975 et moins de 10 dollars en 1980) sur un seul circuit intégré. Il annonçait également que ces circuits permettraient de développer des fonctions logiques aussi bien pour des petits et des gros ordinateurs que pour des appareils qui n'étaient pas des ordinateurs : téléviseurs, calculatrices, machines à écrire, téléphones, jeux, terminaux... Alors que Bruno Lussato travaille sur la question de la décentralisation des organisations, notamment avec François Dalle PDG de l'Oréal, René Dougin, DG de l'entreprise Gloria et d'autres, fortement impressionné, il voit immédiatement que le micro-ordinateur est l'outil de la décentralisation en ce sens qu'il va permettre d'aller vers la déconcentration des moyens informatiques, tandis qu'à ce moment-là les entreprises concentraient de plus en plus leurs flux de données dans des ordinateurs de plus en plus puissants, ce qui avait pour effet de renforcer les tendances centralisatrices des grandes organisations. Nous décidons alors qu'il faut que je me mette sur le champ à travailler sur ce thème et je commence en parallèle un second mémoire d'ingénieur intitulé « La micro-informatique, évolution ou mutation ? ». Je soutiens les deux mémoires en juillet 1973. Durant les vacances, j'entreprends des démarches pour aller faire un PhD à Stanford sur le thème des micro-ordinateurs et lorsque je contacte Bruno Lussato pour une lettre de recommandation, celui-ci me répond « *il n'en est pas question ! je vous prends*

⁴ Ancien bureau du directeur de l'École Centrale des Arts et Manufactures, la salle était devenue le bureau de la chaire de Bruno Lussato. Elle est aujourd'hui une salle des professeurs.

⁵ Première revue d'informatique créée en France, devenue plus tard *L'informatique nouvelle*, puis *L'informatique professionnelle*. La revue avait publié au fil des mois des articles mettant l'ordinateur à la portée de tous. Ces articles furent rassemblés en deux volumes, le premier *Un fil d'Ariane. L'ordinateur à la portée de tous*, signé Marcel Péju en 1977 et le second *Un fil d'Ariane. Microprocesseurs et conception des petits ordinateurs*, signé J.P. Bouhot, Georges Cottin et Jean Tricot en 1979. Ces revues et ouvrages ont été publiés par les Éditions d'Informatique, créées par J.P. Bouhot.

comme assistant au Cnam, nous allons expérimenter les micro-ordinateurs avec la société Gloria qui est d'accord pour essayer et nous écrivons un livre ensemble. Vous m'envoyez le plan dans huit jours ». J'accepte la proposition et nous décidons de rédiger avec Jean-Pierre Bouhot le livre intitulé *La micro-informatique. Introduction aux systèmes répartis* et qui paraîtra aux Éditions d'Informatique en juillet 1974. C'est à la fois un ouvrage technique qui décrit et présente les micro-ordinateurs et un ouvrage d'organisation qui montre comment utiliser ces outils dans l'entreprise. C'est aussi le premier livre au monde rédigé sur les micro-ordinateurs et la micro-informatique, terme dont Bruno Lussato revendiquera l'invention en 1973.

C'est ainsi que j'entre au Cnam en octobre 1973 comme assistant à la chaire de Théorie et systèmes d'organisation, créée pour lui en 1970. TSO : le nom avait été choisi par Bruno Lussato pour marquer le retournement par rapport à l'OST, l'Organisation Scientifique du Travail, enseignée par l'autre chaire d'Organisation tenue par le professeur Jean Gerbier. La chaire de TSO dispensait quatre demi-unités de valeur de cours, deux demi-unités de valeur de TP (Travaux pratiques) et une unité de cours C qui était un séminaire de recherche destiné aux auditeurs qui allaient préparer le diplôme d'ingénieur en organisation. Je partage mon temps entre quatre activités : l'écriture du livre, puis d'articles et de conférences, les exercices dirigés des cours de Bruno Lussato, le suivi de mémoires d'ingénieurs en orga-

nisation du Cnam et en informatique pour l'IIE, et la mise en œuvre de micro-ordinateurs pour la direction commerciale de la société Gloria. Dès la rentrée 1974, j'assure également un cours d'organisation et micro-informatique à l'IIE, ainsi qu'à partir de 1977, le TP B1 du cours de Bruno Lussato et, en alternance avec lui, les séances du cours C d'organisation.

Je fus le seul assistant de Bruno Lussato pendant toute la vie de la chaire de TSO, celui-ci ayant perdu le deuxième poste affecté à la chaire, occupé en fait par une remarquable secrétaire de direction, Nettie von Scherpenzel, mais non titularisable comme assistante. Le Conseil de département avait alors doublement « puni » Bruno Lussato de cet arrangement en supprimant le poste attribué à la chaire et en ne lui attribuant pas de secrétaire.

Durant les années qui suivent, j'enseigne également au Centre d'Études Pratiques en Informatique et Automatique (CEPIA) dépendant de l'Institut de Recherche en Informatique et Automatique (IRIA), à la Maîtrise d'Informatique Appliquée à la Gestion des Entreprises (MIAGE) de Montpellier (qui avait programmé un cours de micro-informatique), à l'ENA, à la MIAGE et à l'Institut Commercial de Nancy avec des cours allant de la micro-informatique à l'organisation du travail et de l'entreprise, en passant par les méthodes d'organisation et de conception de systèmes d'information. Je publie quelques articles dans des revues grand public et anime trente-cinq conférences pour la plupart dans les milieux profes-

sionnels et quelques-unes dans le milieu académique. Je suis également amené à intervenir auprès d'entreprises et d'administrations telles que la Fromagerie des Chaumes, Primagaz, Agfa-Gevaert, Sommer-Allibert, des PME lorraines, La CCI de Meurthe et Moselle, la Direction de la Concurrence et des Prix, La Répression des Fraudes, Saint-Gobain, pour des missions d'audit organisationnel et informatique, de réorganisation et d'implantation des micro-ordinateurs.

Parallèlement aux travaux sur la micro-informatique et la décentralisation, je travaille également sur les méthodes de planification de l'informatique, les schémas directeurs de systèmes d'information, les méthodes de diagnostic des organisations et les méthodes de conception des systèmes d'information. Pour ce dernier domaine, la plupart des méthodes employées ne se fondaient pas sur une étude d'organisation de fond conduisant à raisonner en partant du métier de l'organisation, de ses processus, de sa structure organisationnelle et des personnes qui la constitue, afin d'aboutir à des choix informatiques adaptés à l'organisation, mais conduisaient plutôt à adapter l'organisation à l'informatique. Je ne citerai que la méthode Merise qui était en fait à l'origine une méthode de conception de bases de données érigée en méthode de conception de système d'information et parfois même de schéma directeur !

J'accompagne aussi Bruno Lussato dans ses travaux sur la notion d'information et sur les méthodes de manipulation de

l'information et de désinformation. Il publiera finalement en 2007 un livre intitulé *Virus. Huit leçons sur la désinformation*.

Je veux également souligner un champ important des recherches de Bruno Lussato auquel je fus associé au début des années 1970. Il s'agit des interfaces homme-machine. Bruno Lussato avait développé la notion de degré de synthèse de l'information (France-Lanord, Lussato & Bouhot 1974) et une échelle allant de l'information analytique, « dure », à l'information la plus synthétique, « molle », c'est-à-dire plus déformable. Il insistait sur le fait de bien choisir et d'adapter les outils pour traiter et présenter les informations des différents niveaux. Selon lui, les chambres de contrôle – évoquées plus haut – sont adaptées aux informations de haut niveau de synthèse, les chiffres aux informations analytiques ; alors que des interfaces « iconiques » sont plus adaptées aux êtres humains que les claviers alphanumériques. Il se trouve que c'est ce qui fit le succès de l'interface graphique du Macintosh en 1984. Bruno Lussato proposait en 1973 d'enseigner une discipline qu'il nommait « *graph-ware* » (1980).

De 1978 jusqu'en 1982, Bruno Lussato étant conseil des entreprises Thomson-CSF et Philips pour le développement des lecteurs de vidéodisques optiques qui avaient été inventés par ces deux firmes, je travaillais aussi dans ce domaine. Nous tentions de montrer que le vidéodisque devait être un outil grand public connectable à un micro-ordinateur et visant des applications nouvelles

et interactives. Les deux firmes se fourvoyaient, l'une ayant visé le marché grand public avec un lecteur (Laservision) qui ne permettait pas l'enregistrement et qui venait mal concurrencer les magnétoscopes – avec certes, une qualité d'image et une durée de vie supérieures et un coût inférieur ; et l'autre ayant visé le marché institutionnel avec un lecteur connectable à un micro-ordinateur (TTV), mais d'un coût prohibitif. Les deux expériences se soldèrent par un échec⁶.

En 1979, nous créons au Cnam, avec le Laboratoire de micro-informatique dirigé par Jean Ranchin, la première connexion entre un micro-ordinateur Apple et un lecteur Laservision de Philips. La connexion a été réalisée par un jeune homme de 14 ans, Alexandre Raynaud, aidé par les spécialistes du laboratoire. En 1982, Bruno Lussato publie au Cnam une plaquette destinée à J.P. Chevènement, ministre de la Recherche, intitulée « Pour une politique française du vidéodisque »⁷ ; mais malheureusement l'aventure française des mémoires optiques s'arrêta là. Nous finissons par publier en 1990 un livre dénommé *La vidéomatique. De Gutenberg aux nouvelles technologies de l'information*, analysant les outils de l'information (texte, son et image) et montrant qu'ils allaient tous se retrouver réunis avec la digitalisation et les mémoires optiques.

⁶ Pour plus de détails on pourra se reporter à Lussato & France-Lanord (1990).

⁷ « Pour une politique française du vidéodisque », plaquette éditée par le Cnam, 1982.

En 1983, je soutiens une Thèse de troisième cycle en Sciences de gestion sous la direction de Robert Reix à l'Université de Montpellier avec pour titre « Structure des organisations informatisées : définition de quatre concepts ». Il s'agissait des concepts de centralisation, décentralisation, concentration et déconcentration. Je suis nommé maître de conférences en 1992.

Voyons maintenant ce qu'il en est de la micro-informatique et du livre qui la fonde.

La micro-informatique

Dès le 19 décembre 1973, à l'occasion d'une réunion au Cnam de responsables d'entreprises, Bruno Lussato, Guy Landon (Vice-Président de l'Oréal) et Jean Bounine (conseiller de François Dalle) publient au Cnam avec l'appui de l'Oréal une plaquette intitulée « L'informatique à l'envers », complétée d'une étude de cas au sein de l'entreprise Albert Cochery⁸. Le message est simple : l'expression « à l'envers » caractérise une nouvelle conception du management ou « *taylorisme à l'envers* », une simplification drastique de l'informatique, une technologie avancée permettant la réalisation de micro-ordinateurs polyvalents s'adaptant strictement aux besoins de l'entreprise et évitant une spécialisation

⁸ « L'informatique à l'envers », plaquette éditée par le Cnam, 1973.

excessive du personnel, et une approche méthodologique nouvelle en matière de réorganisation qui permettra réellement la décentralisation au niveau du traitement de l'information et de réduire le « *travail en miettes* »⁹. En permettant le traitement de l'information et la prise de décision au niveau même où l'information est générée, la micro-informatique favorise la rapidité de réponse face à l'événement.

Les dirigeants de l'entreprise de travaux publics Cochery, organisée en une dizaine de centres régionaux, avaient pris conscience des inconvénients de l'informatique centralisée : massification des flux de données et lourdeur des procédures administratives, court-circuitage des régions, duplications de tâches administratives, lenteur d'obtention des informations venant de Paris, avec pour conséquence le maintien d'une comptabilité analytique parallèle dans les centres régionaux. Une étude fut menée pour trouver un matériel et définir des procédures permettant d'effectuer localement les opérations de paye, de saisir au même niveau toutes les autres données et de se donner la possibilité, dans les phases ultérieures, d'effectuer localement d'autres traitements.

Les matériels suivants furent passés en revue : périphériques de saisie qui

maintiennent les postes d'« OS¹⁰ » informatique, mini-ordinateurs et tout récents micro-ordinateurs. Naturellement le choix du micro-ordinateur fut fait, pour un montant amorti en deux ans. Les développements furent rapides grâce au langage Basic et l'équipe informatique très motivée par le fait de penser l'application, la réaliser et la tester aussitôt, directement grâce à l'écran et au clavier. Les « utilisateurs » avaient ainsi l'impression de diriger leur machine et non plus d'être dominés par la logique implacable de l'informatique.

Le livre *La micro-informatique. Introduction aux systèmes répartis*

Le livre part de deux constats : le premier est l'arrivée sur le marché des nouveaux outils, les micro-ordinateurs ; le deuxième est le besoin pour la grande majorité des entreprises de décentraliser et de diversifier leur structure.

Jusqu'à ce moment-là, l'élément « noble et coûteux » de l'ordinateur était l'unité centrale (environ 75 % du coût). On avait alors adopté comme unité d'appréciation d'une machine le coût d'exécution d'une instruction et bien vite un effet d'échelle significatif se manifesta : plus l'ordinateur était puissant (donc coûteux) plus le coût d'exécution d'une instruction était faible. Cette relation

⁹ En référence à l'ouvrage éponyme de Georges Friedmann (1956) qui présente l'étude de la division des actes de production induites par la spécialisation des tâches industrielles et qui montre que l'être humain n'est pas adapté à de telles tâches.

¹⁰ Ouvrier Spécialisé.

fut dénommée loi de Grosch¹¹ du nom de celui qui le premier l'avait formulée, sous cette forme : « *la productivité d'un système d'ordinateur est proportionnelle au carré de son coût* »¹². Cette loi eut un impact considérable et contribua à concentrer les traitements informatiques au sein de machines de plus en plus puissantes. Jean-Pierre Bouhot, qui connaissait Herbert Grosch, constatant l'évolution des circuits intégrés et des mémoires, l'évolution rapide des mini-ordinateurs et l'arrivée des micro-ordinateurs annonce alors qu'aujourd'hui la ressource coûteuse, ce sont les périphériques, dont le prix augmente avec les performances, et pas l'unité centrale. Par ailleurs, il montre que remplacer des traitements en multi-programmation dans un ordinateur central par des traitements répartis entre des machines mono-programmées conduit à des programmes beaucoup plus simples et coûte moins cher. Il énonce alors la « loi de Bouhot » : « *le coût le plus bas d'un travail de gestion est obtenu avec le plus petit ordinateur (mono-programmé) acceptable* ».

Puis, partant du second constat d'une évolution des besoins des entreprises pour plus de décentralisation, on peut tenter de définir l'informatique répartie.

¹¹ Herbert R.J. Grosch (1918-2010) est un brillant pionnier de l'informatique. Il travailla chez IBM dès 1945 et a formulé la loi en 1950.

¹² Citée par J.P. Bouhot dans le livre *La micro-informatique* (1974, p. 16).

La voie pour aller vers une réelle décentralisation (voie *Bottom-up*) consiste à partir du concept de cellule ou de poste de travail. Au lieu de concevoir une organisation fondée sur une hiérarchie d'autorité, on abaisse le point de responsabilité au niveau le plus proche de l'action où l'information est suffisante pour justifier cette responsabilité. Les décisions sont prises par celui qui a les moyens de les prendre, c'est-à-dire qui possède l'information nécessaire et la responsabilité. Les principales questions informationnelles qui se posent alors sont :

- La consolidation des informations nécessaires au sommet. Seules les informations destinées à s'agréger doivent remonter.
- La définition des exceptions. Les informations à caractère exceptionnel peuvent être traitées manuellement ou par des moyens plus lents.
- L'homogénéité. Toutes les informations destinées à être remontées doivent être homogènes.
- Le stockage local des données locales.
- Les applications particulièrement sophistiquées sont traitées au siège ou sous-traitées.

Nous définissons alors l'informatique répartie à partir des caractéristiques suivantes :

1. Volumes faibles au niveau d'une cellule

Les volumes d'informations à traiter au niveau d'une cellule sont très faibles

par rapport à ceux traités dans un système centralisé.

2. Temps d'accès raisonnables

Les volumes restant faibles, le temps d'accès à une information, de même que le temps de réponse de la machine n'ont plus qu'une importance relative.

3. Traitements simples

Les traitements effectués au niveau décentralisé sont toujours beaucoup plus simples que ceux effectués au niveau central. Plus la cellule est proche du terrain, plus elle a un travail spécifique ; elle n'a donc pas à tenir constamment compte de l'ensemble des cas de figure possibles.

4. Outils conviviaux

C'est à Ivan Illich que nous devons l'application de ce concept à la technique. J'étais tombé par hasard sur un article du journal écologiste *La gueule ouverte* qui évoquait la sortie de son livre *La convivialité* (1973) et avais vu d'emblée que c'était le concept qu'il nous fallait pour caractériser l'informatique répartie. On parle d'outils conviviaux, c'est-à-dire restant à échelle humaine. « *L'outil est convivial dans la mesure où chacun peut l'utiliser, sans difficulté, aussi souvent ou aussi rarement qu'il le désire, à des fins qu'il détermine lui-même. Personne n'a besoin d'un diplôme pour avoir le droit de s'en servir ; on peut le prendre ou non.* » (Illich, 1973, p. 45).

5. Faibles coûts

Les volumes restant faibles dans chaque cellule, les performances des outils n'ont pas besoin d'être considérables. Par ailleurs le coût des outils qui apparaissent va baisser fortement dès qu'ils seront produits en plus grande quantité (loi d'économie d'échelle).

6. Niveaux hiérarchiques

Il s'agit, lors de la définition des besoins d'automatisation, de procéder à une analyse différentielle qui implique une hiérarchisation des temps de réponses et des niveaux de complexité. On s'efforce de lutter contre la tendance à l'unification des solutions informatiques qui entraîne un alignement des besoins sur les contraintes les plus sévères et l'utilisation fréquente de l'argument « qui peut le plus peut le moins ».

7. Cohésion des cellules

L'autonomie recherchée dans les structures cellulaires¹³ appelle en contrepartie une cohésion rigoureuse ; ceci met l'accent sur les moyens de communication entre les cellules qui peuvent s'effectuer de différentes manières.

¹³ Bruno Lussato avait publié au Cnam en 1970 un document qui, partant du constat des conséquences négatives des structures centralisées, proposait des structures décentralisées dont la cellule qu'il définit est l'unité de base. « Définition et organisation des structures cellulaires », document édité par le Cnam (1970).

Ensuite nous proposons une définition de ce que devrait être l'outil de l'informatique répartie, le micro-ordinateur, à partir des caractéristiques suivantes :

1. Modularité

Bruno Lussato avait déjà beaucoup travaillé sur la notion de modularité¹⁴, en particulier avec des fabricants de caisses enregistreuses. Il nous est apparu qu'une caractéristique importante des outils de la micro-informatique devait être la modularité, c'est-à-dire la possibilité de modifier, d'ajouter ou de supprimer certains de ses éléments sans que son fonctionnement global en soit affecté.

2. Faible coût

3. Transparence

La notion de transparence s'applique ici à trois niveaux différents :

- Le langage de programmation : plus le langage est transparent plus il est proche des utilisateurs.
- Le clavier : la transparence se traduit par l'adéquation du clavier aux opérations à effectuer.
- Le guidage programmé : c'est l'application des techniques d'enseignement programmé aux micro-ordinateurs et l'utilisation de claviers « programmés » et d'écrans équipés de tablette à dessin et de photostyle, par exemple.

La transparence s'apparente à la convivialité.

4. Performances adaptées aux besoins

Les micro-ordinateurs ne sont pas conçus pour traiter des chaînes complexes de programmes sophistiqués, mais pour résoudre des problèmes simples et spécifiques : une facturation, une gestion de stocks, une comptabilité client...

5. Autonomie

Nous distinguons :

- L'autonomie physique : les micro-ordinateurs sont des ordinateurs à part entière et ne sont pas assimilables à des terminaux, ce qui n'exclut pas de les « connecter » à d'autres ordinateurs en particulier pour transmettre des fichiers.
- L'autonomie du software : cette notion découle de la transparence. Il s'agit de donner à l'utilisateur la possibilité de modifier la programmation de son micro-ordinateur à un niveau donné, bien entendu sous contrôle.

6. Taille réduite

Les micro-ordinateurs doivent être facilement transportables.

7. Maintenance réduite

La maintenance des machines représentait un coût important des gros systèmes, celui-ci étant généralement calculé

¹⁴ Voir l'article « La modularité » de Jean-Xavier Scieller (1971).

sur la base d'un pourcentage (environ 10 %) du coût de location. Les micro-ordinateurs devraient pouvoir fonctionner sans contrat de maintenance, plusieurs facteurs pouvant concourir à cela : fiabilité des composants LSI¹⁵, réduction des parties mécaniques des périphériques, modularité du hardware permettant des échanges-standard en cas de panne...

Le livre présente également les cinq premiers micro-ordinateurs présents sur le marché à la fin de 1973, dont trois Français : Micral¹⁶ de la société R.E.E, Gessix 1000 (élaboré à partir du Micral), Alvan¹⁷, Hewlett-Packard 9830 et Wang 2200.

Vient ensuite un chapitre consacré à l'analyse des facteurs qui vont freiner le développement de la micro-informatique. En effet, nous le percevons d'emblée, différents facteurs se combinaient pour freiner le changement radical proposé par la micro-informatique. Le premier consistait en l'hostilité plus ou moins active de ceux qui avaient quelque chose à perdre de sa diffusion. Et ils étaient nombreux : les constructeurs dominants (en particulier IBM que nous appelions JCN, pour « Jumbo Computer Nirvana »), les responsables informatiques qui avaient, pour

nombre d'entre eux, été formés par IBM ou d'autres constructeurs d'ordinateurs, les informaticiens qui ne connaissaient que la programmation en langage procédural sur de gros systèmes fonctionnant par lots sans interactivité, les sociétés de services qui vivaient du développement d'applications pour gros systèmes et la presse informatique qui suivait le mouvement et vivait des annonceurs, tous concernés par les gros systèmes.

Un second facteur résidait dans l'incapacité des fabricants à définir de façon adéquate les besoins des utilisateurs en faisant abstraction de leurs propres préoccupations, essentiellement centrées sur la fabrication et le marché. Un troisième facteur était le fait qu'en 1974, les grands constructeurs commençaient à proposer de passer au télétraitement, c'est-à-dire la connexion des terminaux passifs à l'ordinateur central pour les utilisateurs. Cette solution était présentée comme une évolution vers une forme de décentralisation, alors qu'il s'avéra que ce n'était pas du tout le cas.

D'autres facteurs économiques, psychologiques et sociaux venaient renforcer la résistance au changement. Ce chapitre présente également, sur un mode humoristique, quarante-huit formules-types, les « *International Standard Objections* », que l'on entendait formulées par les détracteurs de la micro-informatique – et les réponses à y apporter.

Le livre traite finalement la question de l'impact du micro-ordinateur sur

15 LSI : Large Scale Integration, technologie d'intégration des composants à grande échelle des micro-processeurs.

16 Le Micral est le premier micro-ordinateur français, inventé par François Gernelle et André Truong Trong Thi et fabriqué par la société Réalisations et Études Électroniques. Cf. Petitgirard (2020).

17 L'ordinateur Alvan, du nom de la société créée par deux frères Alain et Yvan Grébert. Cf. Gréber (2003).

l'organisation de l'entreprise et la création de cette nouvelle façon de faire de l'informatique que nous avons dénommée micro-informatique, en rupture complète avec le courant dominant. Pour cela Bruno Lussato développe différents thèmes et les approfondit sur le plan théorique :

- La nécessaire définition des frontières entre les fins – le management, la transmission de l'information et le traitement des flux de l'entreprise – et les moyens : les méthodes et les outils électroniques ou manuels. Il montre que tout choix d'outil doit être précédé d'une réflexion organisationnelle et que surtout il ne faut jamais confier au spécialiste des outils la conception de l'organisation de l'entreprise.
- La concordance qui existe entre le degré de formalisation de l'information à traiter et le degré de sophistication des outils correspondants. Il répartit les processus d'information en quatre degrés croissants de complexité (processus d'exécution, processus programmés, processus pseudo-programmés et processus psycho-programmés) qui mettent en jeu des informations de niveaux de complexité et des niveaux de synthèse différents. Plus l'on progresse dans le degré de complexité, plus le remplacement de l'humain par la machine devient coûteux et difficile. Il développe aussi une échelle des degrés de synthèse de l'information

à partir de la notion de « *chunk*¹⁸ », présentée par le psychologue américain G.A. Miller dans l'article « The magical number seven, plus or minus two : some limits on our capacity for processing information »¹⁹. En fonction du niveau de synthèse des informations à prendre en compte, l'outil de traitement et de présentation de l'information à utiliser est différent.

Il présente un historique du développement de l'automatisation dans les entreprises, résumé dans le schéma « Évolution des systèmes d'information » (cf. fig. 1, *infra*) :

1^{re} étape : Système primitif, manuel et non organisé. Il a été dépassé à cause de sa faible efficacité, bien que sa flexibilité et la redondance de sa structure engendrent créativité et stabilité.

2^e étape : L'organisation « scientifique » du travail. Simplification du travail, spécialisation verticale et horizontale des tâches et traitement séquentiel de masse. La productivité est accrue au prix d'une perte sensible de flexibilité et d'intérêt vis-à-vis du travail parcellisé.

¹⁸ Pour G.A. Miller, un *chunk* est l'unité mesurant le nombre de « choses » qu'un être humain peut percevoir simultanément, le maximum étant 7 plus ou moins 2. Bruno Lussato constitue son échelle en associant le degré de synthèse de l'information et le nombre de *chunk* qu'elle contient.

¹⁹ Article publié dans *Psychological Review*, n° 63, 1956, pp. 81-97.

3^e étape : La mécanisation. À l'organisation succède la mécanisation de tâches administratives (facturières, calculatrices de bureau, duplicateurs sélectifs d'ordonnancement...). La productivité continue à croître et des tâches fastidieuses sont évitées. La fiabilité est notablement améliorée.

4^e étape : L'informatique. Une nouvelle étape dans l'efficacité, mais aussi dans la rigidité. La production de masse et l'ordinateur (caractérisé d'ailleurs de « central ») impliquent la centralisation des traitements et l'usage de langages spécialisés. Plusieurs spécialistes « en blouse blanche » se consacrent à la communication homme/machine, ce qui conduit rapidement à la professionnalisation des spécialistes, les informaticiens, et à un double divorce entre eux et ceux qui ne le sont pas, et entre processus manuels et automatisés.

5^e étape : L'intégration et le temps réel. L'intégration des traitements et des données commence et va se développer de façon quasi totale grâce au télétraitement et à la multi-programmation. Les êtres humains deviennent de plus en plus de simples *input* et *output* du réseau informatique, bien que le mode « conversationnel » améliore le dialogue homme/machine en couplant les entrées aux sorties (affichage à l'écran). Le divorce entre la décision humaine et le traitement centralisé atteint son plus haut degré. La fiabilité du système diminue ; sa vulnérabilité augmente.

6^e étape : La micro-informatique. Des micro-ordinateurs sont répartis dans l'organisation à travers un réseau. Cette « dissémination sauvage » et l'absence de langage spécialisé réservé aux professionnels de l'informatique conduisent à un système très flexible combinant la créativité à l'efficacité. On obtient une excellente intégration de l'information au niveau du terrain sans perte d'intégration « utile » au niveau du siège (avec de l'information agrégée et filtrée).

Fin 1973 on en était à l'étape 4 et au passage à l'étape 5. Nous proposons de passer le plus vite possible à l'étape 6. Sur le schéma, le sens interdit symbolise la difficulté et le coût pour passer d'une organisation structurée par l'ordinateur central à une organisation « libérée », n'obéissant qu'aux lois organiques de l'organisation. Bien plus que le passage d'une technologie à l'autre, la difficulté principale était et reste le changement culturel pour passer d'une organisation centralisée à une organisation vraiment décentralisée. Il montre enfin que les différents types de structure ont une incidence forte sur le choix des outils et qu'il convient donc d'approfondir la réflexion sur cette question.

Nous ne sommes jamais parvenus, malgré deux tentatives, à publier le livre aux États-Unis, malgré une traduction et une réécriture non achevée avec Howard Lee Morgan de la Wharton School.

À la suite de cet ouvrage, Bruno Lussato publiera deux autres livres aux-

L'A.M.I.

Le Cnam abrite l'association. Dès 1976, elle comptera une centaine de membres²⁰ dont les fabricants de micro-ordinateurs, mais pas les sociétés de services.

Parmi ceux-ci, on peut citer : Alvan, BSN Gervais-Danone, l'Oréal, CII-HB, HP, le CIC, EDF-GDF, Matra, le CIGREF, Gloria S.A., Primagaz, Cochery, R.E.E., la Banque Worms, la Banque Paribas, les vins Nicolas, Sommer-Allibert, la Fromagerie des Chaumes, Absorba-Poron, Euréquip, Nixdorf, Thomson-SEMS, Logabax, la banque CIC, Armand Thiery et Sigrand, le CESA, l'UTC, et des personnalités comme Jean-Pierre Bouhot, Jean-Louis Le Moigne et François de Closets.

Ses buts :

- faire évoluer, compléter, affiner les méthodes et moyens favorisant la décentralisation des responsabilités au sein de toutes organisations tant publiques que privées ;
- exercer une action de normalisation et de simplification des matériels de traitement, de mémorisation et de communication de l'information, ainsi que des systèmes informatiques et du vocabulaire correspondant ;
- constituer un club d'échanges d'expériences et de réflexion sur la décen-

tralisation des organisations visées ci-dessus.

La presse est invitée à chaque réunion des membres, mensuellement. Des commissions sont constituées pour approfondir diverses questions comme les interfaces, la modularité, l'analyse critique de la conception des outils, les études des besoins, le vocabulaire, les langages de programmation²¹, l'évolution et la normalisation des métiers entre les informaticiens, les organisateurs et les utilisateurs et, d'une façon générale, l'informatique décentralisée. Chaque réunion est l'occasion de présenter l'avancement des travaux des commissions et des expériences qui sont menées au sein des entreprises qui ont décidé de se lancer dans l'aventure micro-informatique.

Dès 1974, Bruno Lussato présente un « Rapport préliminaire pour la Direction de la distribution d'EDF-GDF »²² expliquant les apports possibles des nouveaux outils et des organisations décentralisées et proposant une facturière portable constituée de plusieurs modules et destinée à être utilisée par les agents qui relèvent les compteurs pour éditer la facture sur place. Elle ne verra jamais le jour.

²¹ À titre d'exemple, Bruno Lussato avait confié à un élève de l'IIE, Giancarlo Luxardo aujourd'hui chercheur en linguistique au CNRS, une « Étude sur la transparence des langages de programmation » (Rapport de stage de juin à octobre 1977), qui servit de base aux travaux de la Commission.

²² Bruno Lussato. « Rapport préliminaire pour la direction de la distribution d'EDF-GDF », édité par le Cnam 1974.

En 1977, une plaquette signée François Dalle, Jean Bounine et Bruno Lussato est éditée par l'Institut de l'Entreprise : « Le taylorisme à l'envers ». C'est une sorte de bande dessinée réalisée par Bruno Lussato qui remet en cause le modèle taylorien et bureaucratique et, s'appuyant sur les idées de E.F. Schumacher dans *Small is beautiful* en 1973 et celles développées dans *La micro-informatique*, propose un nouveau mode d'organisation et d'utilisation de la technologie informatique tourné bien évidemment vers la décentralisation et la déconcentration.

En 1977 également l'A.M.I. accueille Georges Zimeray qui, le premier, importe des Apple II en France et qui deviendra président de SONOTEC. Son fils créera Saari.

L'Apple II était doté d'une mémoire vive de 4 à 48 K octets, pouvait supporter jusqu'à huit périphériques à la fois et se programmait en Basic. Il coûtait en France de 9 000 à 17 000 francs.

Toujours en 1977, au mois de mai, l'A.M.I. accueille Simon Nora et Alain Minc qui sont en train de rédiger un « Rapport sur l'informatisation de la société » commandité par le président Giscard d'Estaing. Le choix fait par ces deux auteurs d'aller vers la télématique et le Minitel est considéré par les membres de l'A.M.I. comme tourné vers le passé et d'inspiration centralisatrice et technocratique. Les auteurs resteront sourds à nos conseils et le rapport sortira en janvier 1978 ouvrant la voie

à la télématique française, pour le bonheur de France Télécom et des amateurs de rencontres par Minitel ! Jean Bounine et Bruno Lussato publièrent en 1978 un « contre-rapport » intitulé *Télématique ou Privatique ?* À la suite du rapport, un colloque international nommé « Informatique et société » fut lancé en septembre 1979 pour traiter des enjeux politiques, économiques, sociaux et culturels de l'informatique. Il donna lieu à 165 interventions de personnalités françaises et étrangères et fut clôturé par une grande conférence le 28 septembre au palais des Congrès, sorte d'hymne à la révolution informatique, à laquelle assistait l'intelligentsia française. Parmi les quelques orateurs figuraient Valéry Giscard d'Estaing, Daniel Bell (sociologue américain), André Giraud (ministre de l'industrie) et... Ivan Illich. Son allocution fut une magistrale mise en garde (ou claque) à toute l'assistance. Ironiquement, un an après, en 1980, la revue *Ordinateurs* qui annonçait la publication des cinq volumes des Actes du colloque choisissait de redonner la parole à deux des invités : Bruno Lussato et Ivan Illich.

L'A.M.I. poursuit ses activités jusqu'en 1980, jouant le rôle de catalyseur de la diffusion des possibilités offertes par la micro-informatique comme outil de la décentralisation auprès des entreprises, des pouvoirs publics et de la presse.

Pour illustrer plus concrètement ce que représentait la micro-informatique

au milieu des années 70 et au début des années 80 ainsi que les difficultés de son introduction, examinons deux expériences auxquelles j'ai participé : l'une au sein de la société Gloria et une autre à la Direction Générale de la Concurrence et de la Consommation.

L'expérience au sein de la société Gloria

En 1974, parallèlement à mon activité au Cnam, je me retrouve au sein de la société Gloria, rattaché hiérarchiquement au directeur commercial et fonctionnellement au chef du service informatique, avec pour mission de réorganiser l'administration des ventes. La position est délicate, mes deux supérieurs ne s'entendant pas du tout. Gloria, filiale du groupe américain Carnation Foods, produit et vend du lait en conserve et des produits alimentaires pour animaux familiers (Friskies, Gourmet...). Elle dispose d'usines et de huit directions régionales de vente équipées de dépôt. La structure est classique, fonctionnelle et centralisée. Elle est composée d'environ 1 400 personnes. L'informatique est assurée par un ordinateur IBM 370 sans terminaux. La saisie des données se fait au moyen de machines de saisie et les traitements se font par lots quotidiens, hebdomadaires et mensuels. Le service informatique dépend de la direction financière, comme cela est le cas dans la plupart des entreprises. Ce rattachement à la direction financière dans les premiers temps

de l'informatique de gestion va donner un pouvoir considérable à cette direction fonctionnelle, déjà très puissante, et ceci au détriment des directions opérationnelles. Le sentiment fréquent au sein de ces dernières est de se demander pourquoi est-ce la direction financière qui décide de leurs moyens informatiques. Le degré de satisfaction des services fournis est divers, assez bon chez les comptables, médiocre chez les opérationnels, en particulier à la direction commerciale. Je m'installe à la direction régionale des ventes de Paris et analyse la façon de travailler des vendeurs et de l'administration des ventes. Le premier constat est que le passage du traitement manuel au traitement informatique central s'est traduit par une dégradation du fonctionnement de l'administration des ventes. Auparavant les commandes étaient intégralement traitées à la direction régionale par deux personnes compétentes pour l'ensemble du processus (de la prise de commande au règlement de la facture). Les stocks étaient gérés localement. L'informatisation de ce processus a conduit à alourdir le traitement des commandes ainsi : la commande est rédigée sur un bon de commande dont un exemplaire sert de bon de préparation et de livraison. Un autre exemplaire est envoyé au siège pour saisie par le service informatique et traitement (facturation, gestion des stocks, mise à jour des statistiques, entrée en comptabilité clients). Les règlements sont enregistrés au siège. On procède donc par pré-facturation et spécialisation des tâches entre plusieurs

services. Les conséquences en sont des facturations, des stocks et des statistiques de vente faux. L'information qui est saisie ne correspond pas forcément à la commande effectivement reçue par le client du fait des différentes erreurs possibles (stocks, préparation, livraison, casse, saisie). L'idée que je propose est de déconcentrer les traitements commerciaux (administration et statistiques des ventes) dans les directions régionales au moyen de micro-ordinateurs afin de revenir à un traitement complet de l'administration des ventes dans les directions régionales. Bien entendu le service informatique s'y oppose, avec comme argument le fait que les micro-ordinateurs ne seraient pas assez puissants pour absorber les traitements envisagés. Par ailleurs, IBM qui dicte en fait la politique informatique à ses clients, a convaincu le service informatique de passer à un ordinateur plus puissant, équipé de terminaux permettant de raccorder les gestionnaires à l'ordinateur central et de passer du traitement par lots séquentiels au traitement interactif en temps réel. La direction générale ne sachant comment choisir entre les deux voies accepte que je fasse un essai pendant que le service informatique s'oriente vers la solution IBM.

Un ordinateur Alvan est acheté, que j'installe à la direction régionale dans le bureau de l'administration des ventes. Travaillant depuis plusieurs mois avec l'équipe, je conçois les programmes avec les assistantes de vente et programme

directement sur place, ce qui me permet de leur demander au fur et à mesure des développements ce qu'elles en pensent. Au bout de quatre mois, j'ai réalisé un prototype complet que j'ai testé d'une part en jouant moi-même le rôle d'assistant de vente pendant quelques jours et, d'autre part, en le soumettant aux assistantes. J'invite le directeur général, le chef du service informatique et le directeur commercial à une démonstration qui se conclut par l'accord des trois. Le chef du service informatique, ayant vu concrètement le système, lui accorde sa confiance. Et l'on décide d'une généralisation de cette solution. Les conséquences en seront :

- un coût divisé par six environ de l'informatique commerciale,
- la suppression des erreurs,
- un allègement d'environ 40 % de l'utilisation de l'ordinateur central, ce qui permettra de ne pas louer un ordinateur plus puissant à IBM,
- la suppression des postes de travail spécialisés de saisie,
- la création d'une équipe de trois personnes prenant en charge l'informatique au sein de la direction commerciale,
- la réduction considérable du temps entre une demande de modification et sa mise en service effective ;
- la maîtrise complète de l'ensemble des processus d'administration des ventes, de leur informatique et de leurs évolutions par l'équipe de la direction commerciale,
- la motivation des équipes d'administration des ventes grâce à cette solu-

tion qu'elles maîtrisaient et grâce à la proximité de l'équipe informatique commerciale.

La généralisation de la solution se fera en un an au moyen de micro-ordinateurs Micral Bal multi-postes, la société Alvan faisant face à des difficultés financières.

Sur le plan organisationnel, les résultats de cette expérience furent une déconcentration de l'informatique commerciale avec les micro-ordinateurs et une décentralisation de la fonction informatique vers la direction commerciale.

C'est ce type d'expérimentation qui alimenta les séances de l'A.M.I. au Cnam entre les années 1974 et 1980.

Quatre remarques :

- La surprise de l'ingénieur commercial d'IBM qui venait presque chaque semaine rendre visite au service informatique et qui ne prenait pas au sérieux l'expérience que je menais, lorsqu'il fut mis au courant des décisions prises.
- Le refus des employées qui enregistraient les commandes sur des machines de saisie au siège, lorsque leur furent proposés des postes de travail d'assistante de vente, postes plus polyvalents et plus « intéressants », mais demandant plus d'implication et plus fatigant que la simple saisie qui leur permettait d'écouter de la musique, d'échanger quelques propos et surtout de passer la journée de travail sans de multiples problèmes à régler. Il faut ajouter que le salaire proposé pour ces nouveaux postes n'était pas plus élevé que le salaire pour le travail de saisie, ce qui s'explique par le fait, qu'à l'époque, les salaires dans les départements informatiques étaient plus élevés que dans les fonctions administratives.
- La façon de développer les applications avec les utilisateurs, en testant sur-le-champ les développements au fur et à mesure, qui rappelle un peu les récentes méthodes agiles avant l'heure. Il fallait passer des traitements par lots aux traitements interactifs en temps réel et donc inventer ce nouveau mode d'interaction avec les outils informatiques.
- Le constat du fait que l'introduction de l'informatique de gestion dans les grandes entreprises s'est faite par la comptabilité au sein de services informatiques en général dépendant de la direction financière. Les applications telles que l'administration des ventes furent conçues avec une vision comptable plus qu'avec une vision opérationnelle. Ceci a conduit la plupart du temps à une concentration des flux administratifs et une centralisation des responsabilités associées. On a d'abord contraint l'organisation avec les outils informatiques plutôt que d'adapter l'outil informatique à l'organisation.

Une autre expérience : l'informatisation de la Direction Générale de la Concurrence et de la Consommation. 1981-1984²³

Au début de la décennie 1980 la France sort de l'inflation à 18 % qui avait entraîné un contrôle serré des prix par l'État. Les prix qui ont commencé à être libérés sous le gouvernement de Raymond Barre vont continuer à l'être sous le gouvernement de Pierre Mauroy. Le ministre de l'Économie et des Finances, Jacques Delors nomme en 1981, à la tête de la Direction Générale de la Concurrence et de la Consommation (DGCC). Un homme venant de la société Sommer-Allibert, Claude Jouven. Ce dernier découvre, à un moment où la libération des prix impose un suivi important et rapide de leur évolution, que tout le travail des agents de cette direction qui compte 2481 personnes est fait à la main et que les informations sur les prix remontent des 100 Directions départementales vers les Directions Régionales et vers la Centrale par courrier et au mieux par téléphone. Claude Jouven arrive dans un contexte hostile à un homme venant du privé, au changement et à son idée d'introduire l'informatique. Il est également assez réfractaire au respect des procédures administratives. Il décide alors qu'il faut changer les méthodes et les moyens de travail et passer à l'infor-

matique. Connaissant Bruno Lussato, il fait appel à lui pour le conseiller. Bruno Lussato lui indique une petite société de conseil, MBO, pour laquelle j'avais déjà réalisé une mission. En même temps, Claude Jouven fait appel en interne à un chargé de mission auprès du Directeur du personnel qui s'occupait des primes des agents et de l'organisation de la DGCC, Alain Vauthier doté d'une forte personnalité et très ouvert à l'innovation, qu'il charge de la mission d'informatisation.

Pour obtenir des crédits, il faut respecter les procédures et réaliser divers documents dont le plus important sera un premier schéma directeur, s'appuyant sur une étude de l'existant, et réalisé en interne durant l'été 1982 par Alain Vauthier²⁴. L'étude de l'existant montre que les moyens techniques dont dispose la direction centrale pour le traitement de l'information et la communication se bornent à 77 machines à écrire (dont 37 électriques), deux machines à tirer en offset Varitype, 89 Machines à calculer (une pour deux cadres), 3 lecteurs de micro-fiches permettant de consulter 3000 micro-fiches, une table à dessin, 3 copieurs, un atelier d'imprimerie et de photographie, des téléphones et un unique télex. Certaines tâches font l'objet d'un traitement informatisé effectué par un bureau du Ministère : traitement statistique, contrôle et archivage des marges

²³ La relation de cette expérience est fondée sur mes souvenirs, ceux d'Alain Vauthier, acteur de cette informatisation et sur deux documents rédigés par Alain Vauthier issus des Archives du Ministère de l'Économie et des Finances.

²⁴ Alain Vauthier. « Schéma directeur informatique et bureautique de la DGCC ». Ministère de l'Économie, des Finances et du Budget. DGCC. 1982. Archives du Ministère de l'Économie et des Finances. Boîte d'archives, cote B-0074841.

commerciales à partir des 100 000 relevés manuels faits lors des deux enquêtes annuelles, traitement des 2 000 fiches de contentieux mensuelles et traitement du suivi de l'activité des services extérieurs. Parmi les multiples tâches figure la rédaction de textes dont le volume ne cesse de croître.

Parallèlement, je réalise des études de terrain auprès de quelques directions départementales et régionales accompagné par Alain Vauthier qui essaye de mobiliser des agents sur le thème d'un grand projet original, ne faisant pas appel aux technologies lourdes classiques et demandant la participation de tous. La plupart des directeurs départementaux et régionaux y étaient opposés au début. Très vite, dès la fin de 1982, des secrétaires sont formées au traitement de textes à la Centrale. Dans les départements, il est proposé une formation sur un logiciel de traitement de texte, le tableur Visicalc et Profile III, un gestionnaire simple de fichiers, aux agents qui le souhaitent. Les volontaires vont développer eux-mêmes des applications qui leur servent dans leur travail. Ces applications sont ensuite soumises au service informatique central qui les valide. Ainsi se constitue peu à peu un portefeuille d'applications à la disposition de tous. Dans un second temps l'ensemble des agents seront formés à l'utilisation des logiciels par les GRETA.

Deux contraintes pesaient sur le projet. D'une part la lourdeur et la lenteur des procédures de la Commission de développement de l'informatique du minis-

tère : le schéma directeur fait en 1982 sera présenté en janvier 1984²⁵. D'autre part, procéder au développement de l'informatique locale et régionale avec de l'argent venant de la Direction Générale du Ministère des Finances pour acheter des micro-ordinateurs et payer les consultants et les formations des agents.

Une autre difficulté résidait dans le fait d'introduire de l'innovation dans un contexte non préparé et donc hostile.

À partir de 1984 le projet va être officialisé, avec un plan, un budget, la constitution d'une équipe informatique à la Centrale et continuera à se déployer jusqu'en 1985, avec l'implantation progressive de :

- 7 Mini 6/38 de Bull pour les Directions régionales ;
- un Mini 6/48 de Bull, 16 systèmes de bureautique lourde, une dizaine de micro-ordinateurs et un accès aux banques de données juridiques et économiques pour la Centrale ;
- 150 micro-ordinateurs TRS 80 modèle IV Tandy et 200 micro-ordinateurs portables Tandy modèle 100, dotés des logiciels Superscript, Visicalc et Multiplan, Profile III et

²⁵ Alain Vauthier. Note pour le Directeur général. Ministère de l'Économie, des Finances et du Budget. DGCC. 17 Janvier 1984. Archives du Ministère de l'Économie et des Finances. Boîte d'archives, cote B-0074841.

du langage Basic pour les Directions départementales.

La méthode de travail employée fut de constituer des groupes de réflexion autour de l'organisation du travail à la Centrale et dans les directions départementales et de renforcer le dialogue entre le groupe de travail informatique, les organisations syndicales et les services départementaux.

Cette approche déconcentrée et décentralisée de l'informatisation de la DGCC constitua une véritable révolution au sein du Ministère des Finances et présenta divers avantages qui furent peu à peu reconnus :

- le renforcement des moyens de traitement de l'information de la Centrale et la redistribution aux directions départementales des informations locales jusqu'alors centralisées ;
- la rapidité de l'introduction de l'informatique ;
- l'acceptation aisée et l'appropriation par les agents des applications informatiques et des nouvelles méthodes de travail conçues par eux et avec eux ;
- un coût nettement inférieur à toute solution centralisée ;
- la facilité accrue et l'efficacité apportée dans les tâches à tous les niveaux en particulier la rapidité d'action et

l'adaptation à des tâches de plus en plus variées ;

- la maîtrise par la DGCC elle-même de ses outils informatiques.

Conclusion

Après beaucoup d'années passées à essayer de mettre l'informatique au service des organisations et des personnes qui les constituent, et non pas le contraire, je conclus ces quelques pages par une mise en garde qui est une citation d'Ivan Illich dans *La Convivialité* : « *Lorsqu'une activité outillée dépasse un seuil défini par l'échelle ad hoc, elle se retourne d'abord contre sa fin, puis menace de destruction le corps social tout entier* » (1973, p. 11).

Extrait de la bibliographie des cours de Bruno Lussato au début des années 1970

Allport F. (1965). *The theories of perception and the concept of structure*. New-York : John Wiley and sons.

Ansoff I. (1969). « Repenser l'organisation : problèmes de langage », *Prospective et politique*. Paris : OCDE, pp. 333-374.

Ross Ashby W. (1958). *Introduction à la cybernétique*. Paris : Dunod.

Attali J. (1970). *La parole et l'outil*. Paris : PUF.

Beer S. (1966). *Decision and control*. John Wiley and sons.

Beer S. (1967). *Cybernetics and management*. The English university press.

Beer S. (1972). *Brain of the firm*. Allen Lane. Londres : The Penguin press.

von Bertalanffy L. (1968). *General System theory : Foundations, Development, Applications*. New York (trad. française : Paris, Dunod, 1973).

Blake R. & Mouton J. (1964). *The managerial grid*. Gulf publishing.

Bounine-Cabalé J. & Dalle F. (1972). *L'entreprise du futur*. Paris : Calmann-Lévy.

Bonsack F. (1961). *Information, thermodynamique, vie et pensée*. Paris : Gauthier-Villars.

Chomsky N. (1968). *L'étude formelle des langues naturelles*. Paris : Gauthier-Villars.

Couffignal L. (1958). *Les notions de base*. Paris : Gauthier-Villars/Collection Information et cybernétique.

Culberson J.T. (1963). *The mind of robots*. Urbana : University of Illinois press.

Dale E. (1967). *Comment sont organisées les entreprises américaines*. Paris.

Dearden J. (1966). « Myth of real time management information », *HBR*, janvier-février.

Dearden J. (1967). « Computer : no impact on divisional control », *HBR*, janvier-février.

Dearden J. (1972). « MIS is a mirage », *HBR*, janvier-février.

Drucker P. (1970). *La grande mutation ; vers une nouvelle société*. Paris : Éditions d'organisation.

- Emery F.E. (1969). *Systems thinking*. Londres : Penguin books.
- Fayol H. (1916). *Administration industrielle et générale*. Paris : Dunod.
- Festinger L. (1957). *A theory of cognitive dissonance*. Stanford.
- Forrester J. (1961). *Industrial dynamics*. Cambridge (Mass.) : MIT Press.
- Forrester J. (1966). *Urban dynamics*. Pegasus Paperback.
- Forrester J. (1973). *World dynamics*. Wright Allen Press.
- Foucault M. (1966). *Les mots et les choses*. Paris : N.R.F.
- Gélinier O. (1966). *Le secret des structures compétitives*. Paris : Éditions hommes et techniques.
- Greimas A.J. (1966). *Sémantique structurale*. Paris : Larousse.
- Hayakawa S.I. (1949). *Language in thought and action*. Harcourt.
- Jantsch E. (1969). *Prospective et politique*. Paris : OCDE.
- Johnson R.A., Kast F.E. & Rosenzweig J.E. (1962). *The theory and management of systems*. New York : Mac Graw Hill (trad. française : Dunod, 1970).
- Klir J. & Valach M. (1965). *Cybernetic modelling*. Londres : Iliffe Books.
- Korzybski A. (1933). *Science and sanity : an introduction to non-aristotelian systems and general semantics*. The international Non-Aristotelian Library Publishing Company, rééd. 1958.
- Le Moigne J.L. (1973). *Les systèmes d'information dans les organisations*. Paris : PUF.
- Lévi-Strauss C. (1958). *Anthropologie structurale*. Paris : Plon.
- Lewin K. (1964). *Psychologie dynamique*. Paris : PUF.
- Mc Culloch W.S. (1965). *Embodiments of mind*. Cambridge : MIT Press.
- Mc Donough A.M. (1963). *Information economics and management systems*. New-York : Mc Graw Hill.
- March J. G. (ed.) (1965). *Handbook of organization*. Chicago : Rand Mc Nelly and Company.

- March J.G. & Simon H.A. (1971). *Les organisations*. Paris : Dunod.
- Meadows D., Randers J. & Behrens III W.W. (1972). *Halte à la croissance*. Paris : Fayard.
- Moles A. (1973). *Théorie de l'information et perception esthétique*. Paris : Denoël.
- Osgood C. (1957). *The measurement of meaning*. Urbana : The University of Illinois Press.
- Piaget J. (1967). *Logique et connaissance scientifique*. Paris : Encyclopédie de la pléiade.
- Rosenblatt F. (1958). *The perception, a perceiving and recognizing automation*. Buffalo : Cornell Aeronautical laboratory.
- de Rosnay J. (1975). *Le macroscope*. Paris : Le Seuil.
- de Saussure F. (1949). *Cours de linguistique générale*. Paris : Payot.
- Schumacher E.F. (1973). *Small is beautiful. Une société à la mesure de l'homme*. Paris : Seuil.
- Shannon C. & Weaver W. (1949). *The Mathematical Theory of Communication*. Urbana : The University of Illinois Press.
- Simon H.A. (1960). *The new science of management decision*. New-York : Harper & Row.
- Simon H.A. (1947). *Administrative behaviour*. New-York : Mc Millan.
- Simon H.A. (1967). *Models of man*. New-York : John Wiley and sons.
- Vogel T. (1965). *Théorie des systèmes évolutifs*. Paris : Gauthier-Villars.
- Weizenbaum J. (1976). *Computer power and human reason*. San Francisco : W.H. Freeman (trad. française : Paris, Éditions d'informatique, 1981).
- Wiener N. (1948). *Cybernetics, or Control and Communication in the Animal and the Machine*. Hermann Paris : The MIT Press.

Références de l'auteur

France-Lanord B. & Lussato B. (1990). *La vidéomatique. De Gutenberg aux nouvelles technologies de la communication*. Paris : Éditions d'informatique.

Bouhot J.-P., France-Lanord B. & Lussato B. (1974). *La micro-informatique. Introduction aux systèmes répartis*. Paris : Éditions d'informatique.

Bibliographie

Beer S. (1966). *Decision and control*. London : John Wiley and Sons.

Bounine J. & Lussato B. (1978). *Télématique ou privatique ?* Paris : Éditions d'informatique.

Bounine J., Dalle F. & Lussato B. (1977). *Le taylorisme à l'envers*. Paris : Institut de l'Entreprise.

Bouhot J.-P., Cottin G. & Tricot J. (1979). *Un fil d'Ariane. Microprocesseurs et conception des petits ordinateurs*. Tome II. Paris : Éditions d'informatique.

Foster C.C. (1972). « A view of computer architecture ». AFIPS conference proceedings, Vol. 41 part II, Montvale (NJ.) : AFIPS Press.

Friedmann G. (1956). *Le travail en miettes*. Paris : Gallimard.

Grébert Y. (2003). « L'aventure ALVAN : France et micro-informatique, un mélange délectable », Compte-rendu du Colloque international Histoire de la micro-informatique en France et en Europe (organisé à l'occasion du 30^e anniversaire du micro-ordinateur par le Musée des Arts et Métiers. Paris, 10 décembre 2003).

Illich I. (1973). *La convivialité*. Paris : Éditions du Seuil.

Lussato B. (1972). *Introduction critique aux théories d'organisation*. Paris : Dunod.

Lussato B. (1973). « Histoire d'un dinosaure malheureux ». *Informatique et gestion*, n° 52, novembre, p. 85.

Lussato B. (1973-1974). « La structure de l'information de haut niveau ». *L'informatique*, novembre-décembre 1973 / janvier 1974.

Lussato B. & Illich I. (1980). « Les dérangeurs ». *Ordinateurs*, 22 septembre 1980.

Lussato B. (1980). *Le dossier de la micro-informatique*. Paris : Les Éditions d'organisation.

Lussato B. (2007). *Virus. Huit leçons sur la désinformation*. Paris : Édition des Syrtes.

Miller G.A. (1956). « The magical number seven, plus or minus two : some limits on our capacity for processing information ». *Psychological Review*, n° 63, pp. 81-97.

Nora S. & Minc A. (1978). *L'informatisation de la société*. Paris : La Documentation française.

Péju M. (1977). *Un fil d'Ariane. L'ordinateur à la portée de tous*. Tome I. Paris : Éditions d'Informatique.

Petitgirard L. (2020). « Biographie de l'ordinateur R2E-Micral, ou comment faire exister un 'micro-ordinateur' dans les années 1970 » [en ligne]. *Technologie et innovation*, vol. 5, « Biographies d'innovation » [URL : https://www.openscience.fr/IMG/pdf/iste_techinn20v5n2_1.pdf].

Scieller J.-X. « La modularité ». *L'informatique*, janvier 1971.

Schumacher E.F. (1973). *Small is beautiful : A Study Of Economics As If People Mattered*. London : Blond & Briggs.

La naissance de l'informatique à l'École centrale à la fin des années 1960 : un enseignement au service de la formation de managers

Cédric Neumann

HT2S, Cnam

En France, les recherches en sciences sociales portant sur le numérique se sont peu intéressées à l'informatique dite lourde¹ des années 1960-1970. Les travaux abordant la cybernétique (Tricot, 2008 ; Le Roux, 2018) ne se sont pas intéressés à la constitution des ordinateurs comme outil de management opérée par les constructeurs, les entreprises et les administrations utilisatrices. Pour ces acteurs, la transformation d'un instrument de calcul scientifique en outil de management passe par une modification des caractéristiques professionnelles des personnels à son contact. D'une part, les postes de techniciens, en particu-

lier celui d'analyste-programmeur, sont considérés comme des postes de cadres, d'autre part, l'ensemble de la population des cadres doit posséder des compétences informatiques afin de pouvoir dialoguer avec l'informaticien. Ainsi, apparaît dans les politiques d'informatisation une nouvelle catégorie, celle des utilisateurs, définie comme des cadres non spécialisés mais devant être familiers de l'informatique et être capable de s'en servir de manière autonome. Le développement de cette population est considéré comme une condition de réussite de l'informatisation (Neumann, 2013)². La formation et la sensibilisation des cadres à l'informatique ont reposé sur le développement d'enseignements

¹ Ce qualificatif, établi rétrospectivement, désigne l'informatique centralisée reposant sur des grands calculateurs situés dans des centres de calcul. Le qualificatif de lourd oppose cette informatique à la micro-informatique dont sont louées les vertus de légèreté et de décentralisation.

² Cette restriction de la population concernée aux cadres est conforme à la pensée managériale des années 1960. Sur ce point général, cf. Boltanski & Chiapello (1999).

informatiques dans les établissements d'enseignement supérieur, aussi bien les universités que les grandes écoles (Mounier-Kuhn, 2009 ; Neumann, 2013). Dès le dernier tiers des années 1960, l'enseignement de l'informatique, soit fondamentale, soit appliquée à un domaine, se généralise dans toutes les grandes écoles qu'il s'agisse d'écoles de commerce ou d'écoles d'ingénieurs. Le développement soudain d'une matière jusque-là marginale dans les *curricula* de la majorité de ces écoles répond aussi aux craintes d'une pénurie de personnels informatiques par les entreprises et les administrations utilisatrices. La mobilisation de l'appareil de formation a pour fonction de résorber cette pénurie (Neumann, 2013). L'impératif de formation des cadres par l'enseignement supérieur est relayé par la Délégation à l'informatique, créée en 1966, dont une des tâches est de contribuer à définir la politique nationale de formation à l'informatique en impliquant dans celle-ci les Universités et les différentes grandes écoles. Ainsi, en intégrant à leurs *curricula* un enseignement de l'informatique, les grandes écoles démontrent qu'elles sont à « l'écoute du marché » et des besoins de qualification exprimés.

Cependant, le développement de l'informatique dans les grandes écoles ne peut s'expliquer uniquement comme une adaptation nécessaire à des évolutions extérieures sur lesquelles les grandes écoles n'auraient aucune prise. D'une part, en introduisant une nouvelle discipline dans leur cursus, celles-ci

contribuent à modifier l'éventail des compétences professionnelles des cadres et opèrent une redistribution des différents savoirs et manière d'être nécessaires pour devenir cadre. De même, l'importance accordée à cet enseignement – les notions abordées, volume horaire et utilisation de l'informatique comme outil de calcul dans les autres disciplines – est très variable selon les établissements. Elle dépend de l'éventail des disciplines déjà présentes et des débouchés professionnels que l'école peut offrir à ses étudiants. Plus généralement, l'impératif de formation des cadres à l'informatique est retraduit selon la position d'un établissement dans le champ des grandes écoles avec l'objectif de maintenir cette position ou de l'améliorer³.

Je présenterai ici le cas de l'École centrale de Paris à partir d'archives encore largement inexploitées : les Archives de la délégation à l'informatique concernant cette école, produites dans le cadre de la politique nationale de promotion de l'enseignement de l'informatique, les archives conservées au Centre de Documentation de l'École centrale et

³ Cette analyse des grandes écoles en termes de champ est inspirée de Bourdieu (1989). Pour une analyse plus globale des logiques différenciées d'introduction de l'informatique selon les établissements, je me permets de renvoyer à ma thèse : « De la Mécanographie à l'informatique. Les relations entre catégorisation des techniques, groupes professionnels et transformation des savoirs managériaux » (Neumann, 2013, pp. 451-509). Par ailleurs, on peut trouver des analogies entre le développement de l'informatique dans les grandes écoles à la fin des années 1960 et celui de l'internationalisation de celles-ci durant les années 1990. Sur ce point cf. Lazuech (1998).

enfin la revue de l'École, intégralement dépourillée pour la période 1965-1975.

Dans un premier temps, je montrerai que le développement de l'enseignement de l'informatique à l'École centrale s'explique par la volonté, dans un contexte général de réforme des grandes écoles, de préserver son rang d'école généraliste. Cette ambition conduit à placer l'informatique dans les enseignements de gestion plutôt que d'en faire une nouvelle science industrielle liée à des techniques de production. Ensuite j'établirai que ce lien entre l'informatique et la gestion peut être rapporté à l'influence du groupe Centrale-informatique, une association d'anciens élèves de l'École disposant de positions influentes dans le secteur informatique. Puis j'analyserai comment l'informatique s'inscrit dans le *curriculum* de l'École centrale comme discipline de gestion⁴. Enfin, je montrerai que la conversion managériale de l'École entraîne de vives résistances de la part des étudiants. En effet, ces derniers contestent la valeur scientifique des différents enseignements de gestion, dont l'informatique, et leur légitimité à être enseignés à l'École centrale.

Un contexte général de réforme des grandes écoles

Moderniser l'enseignement de l'École centrale

Pour comprendre comment l'enseignement de l'informatique s'est généralisé à l'École centrale, il faut revenir brièvement sur les critiques dont celle-ci et plus généralement les grandes écoles font l'objet durant les années 1960. Cette critique repose sur deux arguments complémentaires : d'une part l'enseignement scientifique donné est démodé et trop spécialisé ; d'autre part, les cursus de ces écoles ne favoriseraient pas l'apprentissage de l'autonomie permettant une intégration rapide aux entreprises. La réforme du cursus des grandes écoles doit donc faire advenir un enseignement qui soit d'un niveau scientifique plus élevé tout en étant plus pratique. En quelque sorte, les grandes écoles doivent favoriser une socialisation plus précoce de leurs élèves aux fonctions de cadre. Une interview du directeur de l'école centrale donnée à la revue *Euroscopie* en 1971 illustre ce projet :

Nous cherchons pour nos élèves qui sortent des classes préparatoires qui sont très contraignantes par la pression du travail, par la hantise du concours, et qui donnent un état d'esprit très scolaire, très potache, à les mettre dans l'état d'esprit qu'ils devront avoir dans l'entreprise. C'est en effet à l'école de prendre le risque du passage du régime d'autorité au régime de responsabilité, et donc de faire comprendre aux élèves

⁴Pour une mise au point sur la sociologie du *curriculum*, cf. Fourquin (2008).

que c'est à eux d'organiser leur travail, de venir ou de ne pas venir aux cours, de suivre ou non des cours à l'extérieur.⁵

Cette volonté de responsabilisation des élèves des grandes écoles passe par une redéfinition des caractéristiques morales des cadres et des ingénieurs. Il s'agit d'adapter ceux-ci et leurs formations à des organisations jugées en perpétuelle évolution. Par exemple, en 1968, une journée d'étude organisée par le Conseil National des Ingénieurs Français (CNIF) propose une « *nouvelle définition du savoir* » car « aujourd'hui il faut comprendre le « *devenir* » et « non plus apprendre pour savoir le « *passé* » »⁶. Désormais, les cadres se définissent par leur polyvalence, leur adaptabilité, rendues possibles par la formation continue, et leur imagination : « *“l'âge des structures en mouvement” a commencé [et] implique l'ouverture permanente de l'esprit de l'ingénieur et l'obligation dans laquelle, il se trouvera de s'informer, de s'instruire, de se former continûment.* »⁷

Un rapport de synthèse de la commission pédagogique de l'École centrale va dans le même sens :

L'époque actuelle se caractérise par une très grande mobilité dans tous les domaines et par une évolution des techniques telles que l'ingénieur se trouvera de plus en plus déchargé des tâches routinières au profit des fonctions créatrices allant de la recherche à l'invention.

Les qualités requises au premier chef chez un ingénieur sont, dans un tel contexte, la combativité, la créativité, l'esprit d'initiative et de décision, l'adaptabilité. Une certaine polyvalence et un sens des relations humaines ne sont pas moins indispensables⁸.

La généralisation de l'enseignement de l'informatique dans les grandes écoles trouve son origine dans la capacité supposée de la nouvelle discipline à répondre aux enjeux scientifiques et normatifs de la modernisation des grandes écoles. En tant que technique récente, elle symbolise l'adaptation de l'enseignement scientifique des écoles aux besoins des entreprises⁹. De même, elle permet de donner une consistance concrète aux aspects normatifs de la définition du cadre généraliste c'est-à-dire la polyvalence, la mobilité, l'adaptabilité et le renouvellement permanent des compétences. Dès 1962, Maurice Teper, directeur

5 Anonyme (1971), « Interview de Baron », *Euroscopie*, 12, p. 17.

6 Archives de l'École centrale, 9.2.1 – Société des ingénieurs civils de France, Commission du perfectionnement continu, *La formation continue*, Journées nationales d'études des Ingénieurs français, 15, 16 novembre 1968 organisées par le Conseil National des Ingénieurs français (CNIF), p. 2.

7 Boucheron R. (1967), « L'école centrale originalité et évolution de son enseignement. », *Hommes et Manufactures*, 179, p 8.

8 Archives de l'École centrale 12.1.1. *Rapport de synthèse sur les travaux de la commission pédagogique*, 1971-1972, 21 février 1972, p. 2.

9 Voir par exemple, Archives de l'École centrale, 10.2.1 – M. David, président du Centre national des jeunes cadres, « Conséquences de la modification du rôle des cadres sur la formation dans les grandes écoles » in Fédération nationale des associations d'élèves en grandes écoles, *Avenir et évolution des grandes écoles d'ingénieurs*, FNAGE, Colloque de Lyon-8-9 mars 1968 p. 71.

technique de l'organisation Paul Planus, un cabinet de conseil, qui enseigne durant les années 1960 au Centre de Préparation aux Affaires de la Chambre de Commerce et d'Industrie de Paris (CCIP) ainsi qu'à l'École centrale associe la « *recherche opérationnelle* », le « *traitement automatisé de l'information* » à « *l'expérience, l'intuition, l'imagination créatrice* »¹⁰. Un congrès, tenu en 1965 à l'Institut technique de l'Illinois et rassemblant plusieurs dirigeants de grandes écoles françaises¹¹, lie explicitement l'utilisation des ordinateurs, la part croissante des sciences dans le travail des cadres et l'adaptabilité de ceux-ci :

L'ingénieur, qui assume des responsabilités accrues, travaille dans un mode d'une grande complexité technique dont les besoins se renouvellent sans cesse.

L'époque actuelle se caractérise aussi par une pénétration de plus en plus grande de la science dans la technique ; la manifestation la plus évidente en est l'emploi généralisé des méthodes d'analyse mathématiques pour résoudre bien des problèmes du domaine de l'ingénieur, avec un recours de plus en plus fréquent aux ordinateurs, au lieu des procédés empiriques souvent utilisés dans le passé.

La conséquence en est un accroissement très sensible du niveau des capacités demandées à un ingénieur. [...] C'est que l'ingénieur devra en effet affronter des problèmes qui ne peuvent être prévus aujourd'hui ; il lui faudra demeurer réceptif à de nouveaux développements tant de la science que des techniques et s'adapter continuellement à de nouvelles situations.

Les qualités jugées essentielles à notre époque sont l'adaptabilité, l'attitude à transmettre sa pensée et à travailler en équipe, la capacité de concevoir de nouvelles machines ou de nouveaux procédés ; plus que jamais l'ingénieur doit être un créateur¹².

L'informatique constitue une opportunité de renouveler l'enseignement de l'École centrale dans le sens d'une plus grande adaptation aux exigences des entreprises. Dans le même temps, la valeur accordée à l'informatique témoigne de la croyance de sa direction dans la réalité des effets sociaux de la « révolution informatique », en particulier sur l'association de l'informatique aux activités de direction ainsi que la polyvalence qu'elle exigerait.

Maintenir son rang d'école généraliste

La modernisation de l'enseignement de l'École centrale répond à l'objectif plus précis de maintenir son caractère généraliste. Ce dernier est régulièrement

¹⁰ Archives de la CCIP. 176 W 58. « Journées nationales d'enseignement supérieur commercial : Évolution des méthodes et des carrières de Direction », Résumé de la communication de Maurice Teper, 15 mai 1962 p. 1-2.

¹¹ On trouve à ce congrès, entre autres, P. Olmer, directeur de l'École Supérieure d'Électricité, Forestier, Directeur de l'École Nationale Supérieure de Chimie de Strasbourg, Bonvalet, Directeur de l'Institut des Sciences de l'Ingénieur de Nancy et Muel, adjoint au Directeur des études de l'École Polytechnique.

¹² Archives de l'École centrale. 12.1.1. Congrès à l'Institut technologique de l'Illinois, 21-25 juin 1965, p. 2.

répété durant les années 1960 comme un rappel de la vocation originelle de l'École. Par exemple, Léger indique que pour former des ingénieurs de niveau élevé « *l'enseignement doit être marqué par le plus grand éclectisme* » et qu'« *il y a une formation générale industrielle, comme une formation générale littéraire ou une formation générale scientifique. Il y a même une seule et unique formation générale qui n'est autre que la culture* »¹³. De même, Boucheron rappelle que le centralien doit « *recevoir une formation générale industrielle qui est la marque originale de notre école* »¹⁴.

Le développement de l'enseignement de l'informatique de l'École centrale est un des aspects de la modernisation de son cursus afin de maintenir sa vocation généraliste et conserver l'écart qui la sépare des petites et moyennes écoles aux cursus plus spécialisés¹⁵. Par exemple, le Comité d'enseignement de l'École centrale fait de l'enseignement de l'informatique à l'École une simple manifestation de cette vocation : « *l'ÉCOLE CENTRALE = école polyvalente = école informatique*¹⁶ ». Le caractère généraliste de cet enseignement détermine la forme que va prendre celui-ci.

L'informatique de gestion comme discipline d'enseignement général

En effet, à partir de la seconde moitié des années 1960 l'informatique apparaît comme un élément de formation générale pour la direction de l'École centrale. L'enseignement de cette discipline permet d'assurer la polyvalence qui définit le cadre généraliste. Lorsque ses professeurs et ses dirigeants justifient la mise en place de l'enseignement de l'informatique, ils se réfèrent à la nécessité de maintenir des débouchés de haut niveau pour leurs étudiants. En 1968, J.-P. Bouhot, un des professeurs d'informatique de l'École, cosigne un article dans lequel il développe cette idée, l'argumentation reposant sur les thématiques principales de la « révolution informatique »¹⁷ :

Même si l'on n'est pas directement concerné et si l'on se sent la conscience tranquille, il est pratiquement impossible d'ouvrir un hebdomadaire sans avoir l'impression que notre société est en train de subir des changements profonds. Ces changements profonds, qui façonnent notre quotidien et l'avenir, semblent se polariser autour d'un concept nouveau :

¹³ Archives de l'École centrale. 7.6.1. J-E Léger, « L'enseignement à l'école centrale », 19/6/67, p. 3.

¹⁴ Boucheron R. (1967), « L'École centrale originalité et évolution de son enseignement », *Arts et Manufactures*, 179, p. 7.

¹⁵ Sur ce point, cf. Bourdieu (1989).

¹⁶ Archives nationales de Fontainebleau 19820263. Art 2. Comité d'enseignement, 19 janvier 1968, p. 1.

¹⁷ Dans les années 1960, les sources managériales évoquent systématiquement une « révolution informatique » qui bouleverserait le fonctionnement des organisations par la libre circulation de celle-ci et l'optimisation de son traitement. Cette « révolution informatique » conduirait à remettre en cause les relations de pouvoir et imposerait le renouvellement permanent des compétences et des qualifications. Par exemple, le colloque de Cerisy de 1970 est intitulé « Révolutions informatiques ». Pour une analyse d'ensemble de ces sources, cf. Neumann (2008, 2013 et 2022).

l'informatique. L'informatique qui pénètre dans notre vie un peu comme l'électricité au début du siècle, modifiera l'enseignement, la gestion des entreprises, la recherche scientifique, les méthodes de production, la médecine et j'en passe.

L'informatique c'est donc l'avenir ; c'est vers l'informatique que les jeunes (et les moins jeunes) doivent s'orienter s'ils désirent rester dans le vent et conserver un potentiel professionnel¹⁸.

Lors de la réunion constitutive du groupement professionnel Centrale-informatique¹⁹, J. Maisonrouge, P.-D.G. d'IBM et ancien centralien va plus loin dans son analyse. D'après lui, l'acquisition de compétences informatiques est la meilleure réponse aux éventuelles craintes de déclassement des centraliens :

Le premier [sujet] concerne l'avenir de notre École. J'ai été surpris dans les deux dernières années, au cours d'entretiens que j'ai eu l'occasion d'avoir avec des Centraux, en particulier dans le cadre de du Comité de l'association, de voir combien nos camarades se plaignaient de ce que l'école n'occupe plus dans l'industrie ni dans le gouvernement, la position qu'elle avait il y a cinquante ans. Les réactions de

nos camarades sont diverses. Certains disent « *toutes les places sont réservées aux Polytechniciens, ou aux gens de l'Ena...* », « *nous ne pouvons pas faire grand-chose* », d'autres disent « *il faut que l'École reprenne la place qu'elle avait dans l'industrie française, et pour cela il faut que nous arrivions à renforcer notre position dans le gouvernement* ». Je crois qu'il est possible de le faire, mais il est important que nous nous rendions compte que cela dépend essentiellement de nous. [...]

Ce qui est important c'est de réaliser que le P.-D.G. de l'avenir, sera un monsieur qui comprendra l'informatique, qui saura ce qu'est un programme, ce qu'est un ordinateur. Si nous voulons donc que les ingénieurs de notre école, prennent dans l'industrie une place au moins égale à celle qu'avaient leurs anciens, il est important qu'ils reçoivent une formation profonde dans le traitement de l'information, de façon à être prêt dès le début de leur carrière, à participer pleinement à la gestion intégrée des entreprises²⁰.

À partir de la deuxième moitié des années 1960, l'informatique n'apparaît pas comme une nouvelle spécialité mais comme une discipline d'enseignement général, une composante du socle de culture générale des diplômés de l'École centrale. L'importance accordée à l'informatique dans la formation du cadre généraliste doit être reliée à une

¹⁸ Bouhot J.-P. et Margulici L., « Informatique et marché du travail », *Arts et Manufactures*, 190, 1968, p. 25. Bouhot est directeur général adjoint de la société de conseil Diebold France et Margulici est ingénieur en chef dans la même compagnie. Les deux auteurs sont d'anciens centraliens.

¹⁹ Nous revenons plus loin sur ce qu'est Centrale-Informatique et son importance dans le développement de l'enseignement de l'informatique à l'École centrale.

²⁰ Archives Nationales de Fontainebleau. 19820263 – Art 2. Discours tenu par Maisonrouge lors de l'assemblée constitutive du groupement Centrale-informatique, 8 février 1968, p. 2-3.

autre évolution : l'émergence de l'enseignement de la gestion au sein des établissements d'enseignement supérieur²¹. L'intégration de différentes techniques au sein d'une « discipline », la gestion, et l'enseignement de celle-ci au sein des établissements d'enseignement supérieur témoigne de la croyance dans l'existence d'un « *management gap* » entre la France et les États-Unis²² et dans le fait que la maîtrise des différents aspects de la gestion constitue l'essence professionnelle du cadre supérieur. Le développement de l'informatique à l'École centrale est le corollaire de l'importance accrue accordée à la gestion et au management. En effet, à partir de 1967-1968, c'est l'informatique de gestion qui est enseignée dans le tronc commun suivi par tous les étudiants de l'école lors des deux premières années.

Les acteurs de l'introduction de l'enseignement de l'informatique dans l'École centrale

Les professeurs de l'École centrale et l'informatique

La promotion de l'enseignement de l'informatique à l'École centrale est assurée, d'une part, par des gestion-

naires ; d'autre part, par des professeurs issus de disciplines industrielles. Parmi les premiers, Maurice Teper – Directeur Général Adjoint de la Compagnie des Machines Bull au début des années 1960, il enseigne les sciences de gestion à l'École centrale ainsi qu'au Centre de Préparation aux Affaires et à l'École d'Organisation Scientifique du Travail – ou Nicolas Manson, ingénieur spécialisé dans les méthodes de Recherche Opérationnelle à IBM qui devient chargé de conférences vacataire en informatique à l'École centrale à partir de 1966. Parmi les seconds, Roger Kling, qui enseigne la thermodynamique et la mécanique des fluides à l'École centrale depuis 1949, est membre du conseil de perfectionnement et du Conseil d'administration de l'École puis devient délégué du département d'informatique en 1972²³. Ainsi, lors de l'année scolaire 1966-1967, l'informatique n'est enseignée que pendant la deuxième année d'études et fait partie des « sciences appliquées à l'art de l'ingénieur » comme la chimie appliquée et le génie chimique, la métallurgie, les constructions mécaniques et les procédés généraux de construction²⁴. Enfin, l'électronique et l'automatisation figurent parmi les disciplines enseignées à l'École centrale dans la première moitié des années 1960 et ont pu fournir les bases pour un

²¹ Sur le développement de l'enseignement de la gestion en France, cf. Pavis (2003) et Chessel & Pavis (2001).

²² Sur ce point, cf. Boltanski (1982, pp. 157-236).

²³ Archives de l'École centrale, 16.1.1. « Compte rendu de la séance du conseil de l'enseignement », 26 Février 1972, p. 1.

²⁴ Boucheron R. (1967), « L'École centrale. Originalité et évolution de son enseignement », Arts et Manufactures, 179, p. 8.

enseignement d'informatique orienté vers le matériel (*hardware*).

Malgré cette diversité initiale des enseignants et des disciplines pouvant contribuer au développement de l'informatique à l'École centrale, dès 1967-1968, l'enseignement de l'informatique à centrale est orienté vers les applications de gestion et concerne les deux premières années du cursus. Cette orientation s'explique par la vision de l'enseignement de l'informatique défendue par le groupement professionnel Centrale-informatique pour lequel les débouchés les plus rentables de l'informatique se situent dans le domaine de la gestion. Ainsi, un article du groupement inscrit l'enseignement de l'informatique à l'École centrale dans l'objectif plus général de familiariser les centraliens à la gestion et de combler le fossé présumé entre l'enseignement français du management et les pratiques américaines :

Pendant longtemps, en France, on a estimé que la capacité de diriger des entreprises, le fameux management, était un don merveilleux réservé à une élite capable de dégager de la technique et de l'expérience professionnelle toutes les qualités du parfait P.-D.G. Les responsables de l'enseignement en France ont enfin compris que cette science devait être enseignée aux étudiants. Mais combien d'années après les Américains ?²⁵

²⁵ Brouchet A. (1968), « Centrale-Informatique », *Arts et Manufactures*, 190.

Un lobby au service de la promotion de l'enseignement de l'informatique : le Groupe Centrale-Informatique

Pour comprendre comment l'informatique a pu s'établir rapidement et durablement à l'École centrale, il ne faut pas tenir compte uniquement du rôle des enseignants de disciplines préexistantes mais aussi du réseau de ses anciens élèves présents au sein de l'industrie informatique et dans les directions informatiques de grandes entreprises et d'administrations publiques. Lorsque l'École débute son enseignement de l'informatique en 1967, certains de ses anciens élèves occupent des positions dominantes dans le domaine du traitement de l'information. La promotion de l'informatique à l'École trouve en J. Maisonrouge un puissant relais. Celui-ci, à sa sortie de l'École centrale en 1948, reçoit une bourse pour étudier les domaines d'application de l'électronique aux calculateurs. Grâce à celle-ci, il se rend aux États-Unis suivre un stage de formation chez IBM au sein duquel il effectuera une longue carrière de dirigeant, en devenant notamment président d'IBM Europe en 1964²⁶. Parallèlement à ses fonctions de dirigeant d'IBM, Maisonrouge conserve un rôle actif au sein de l'École centrale, président des centraux USA, défenseur des thèses

²⁶ Cf. Harris & Sédouy (1977), particulièrement pp. 309-312. Par ailleurs, la carrière de Maisonrouge est intégralement retracée sur le site web d'IBM [URL : http://www-03.ibm.com/ibm/history/exhibits/builders/builders_maisonrouge.html].

du *management gap*²⁷ et de la « révolution informatique », il est, logiquement, l'apôtre du développement de la gestion et de l'informatique dans les enseignements de l'École. Maisonrouge joue un rôle déterminant dans la création du groupement professionnel Centrale-informatique qui rassemble d'anciens centraliens travaillant dans le domaine de l'informatique. En mars 1968, il est élu président du groupe professionnel Centrale-informatique. Le groupe, qui compte 15 membres en 1968 en compte 395 en 1972²⁸. Ceux-ci occupent des positions professionnelles élevées chez les constructeurs, les grandes sociétés de conseil en informatique, les utilisateurs ou encore à la Délégation à l'informatique (cf. Tableau 1).

Dès ses débuts, le groupe Centrale-informatique se donne pour objectif d'influencer la direction de l'École pour intensifier l'enseignement de l'informatique à centrale :

Il faut que nous nous battions pour que l'informatique prenne une place plus

27 Maisonrouge a acquis, lors de ses séjours aux États-Unis et de sa carrière à IBM, une admiration sans limites pour le mode de management d'IBM qu'il érige en paradigme du modèle américain : « *Je n'ai encore jamais rencontré de gens ayant passé un peu de temps aux États-Unis – au moins six mois – qui, dans l'ensemble, n'admirent pas le système américain... J'ai vu beaucoup de jeunes, même récemment – puisque j'étais président de l'association des "Centraux des États-Unis" jusqu'à mon départ en 1974 – j'ai vu beaucoup de camarades passer, eh bien, ils étaient tous enthousiastes !* », in A. Harris et A de Sédouy, *op. cit.*, p. 314.

28 Centrale-informatique (1972), « Compte rendu de l'assemblée générale », *Arts et Manufactures*, 230, 1972 p. V.

importante dans l'enseignement de l'école. [...] Je propose donc que notre groupe soit extrêmement actif en agissant auprès de la direction de l'École et du Comité de perfectionnement, pour qu'un enseignement de l'informatique soit intensifié²⁹.

En 1969, le Directeur de l'École, Jean-Jacques Baron, appelle le groupement à orienter « *la direction de travail pour l'avenir* »³⁰ de l'École. L'influence du groupe auprès de la direction de l'École se voit encore en 1972 lorsque celui-ci lance « *en accord avec le directeur de l'École, une étude de marché pour aider à préciser le profil de connaissances informatiques attendues par les entreprises utilisant les ingénieurs ECP* »³¹. De même, le groupe publie des articles dans la revue *Arts et Manufactures* où il défend l'intérêt du développement de l'informatique à l'École.

L'action de Centrale-informatique ne se limite pas à intervenir auprès de la Direction de l'École. Parallèlement, le groupe agit directement auprès des élèves pour convaincre ceux-ci de la nécessité de l'enseignement de l'informatique. Par exemple, il publie un bul-

29 Archives Nationales de Fontainebleau. 19820263. Art 2. Discours tenu par Maisonrouge lors de l'assemblée constitutive du groupement Centrale-informatique, 8 février 1968, p. 2-3.

30 Anonyme (1969), « Assemblée Générale du groupement centrale informatique », *Arts et Manufactures*, 197, p. 41.

31 Anonyme. (1972). « Compte rendu de l'assemblée générale ». *Arts et Manufactures*, 230, 1972, p. V.

	Promotions	Fonctions
M. Arnaud	1955	Directeur de la division application de CEGOS-Informatique
A. Brouchet	1959	Chef du service ordinateurs et systèmes de la sélection du <i>Reader's digest France</i>
M. Clément	1951	Ingénieur de recherches à l'IRIA, chef de travaux pratiques à l'École centrale
M. Delamarre	1961	Chargé de mission auprès du délégué à l'informatique
A. Hamel	1954	Département de calcul électronique. Centre de recherches nucléaires de Saclay
J.-C. Hirel	1960	Responsable du Centre de calcul de l'Institut de Physique Nucléaire
L.-N. Joly	1961	Directeur de l'assistance technique d'UNIVAC France
G. Lecœur	1935	Ingénieur conseil auprès des tribunaux
D. Leduc	1961	Ingénieur-analyste Control Data
J. Lemonnier	1951	Directeur commercial IBM-France
P. Mathelot	1955	Directeur du développement et de la promotion de la SIA
J. Maisonrouge	1948	Président d'IBM World Trade Corporation
J.-P. Panaget	1961	Représentant pour la France et les pays francophones Excepta Medica Foundation.
P. Raimondo	1959	Ingénieur Citroën
J.-C. Vernières	1958	Contrôleur gestion du département technique de BGE

Tableau 1 - Membres du comité Centrale-informatique en 1968

(Il s'agit d'une reproduction exacte du tableau conservé dans les archives de Fontainebleau. L'ordre dans lequel sont énumérés les membres et leur qualité professionnelle sont ceux indiqués dans le document original.)

Source : « Assemblée constitutive du groupement Centrale-informatique », 8 février 1968 in CAC de Fontainebleau 19820263 Art 2.

letin, intitulé *C.I.B.L.E.*³² destiné aux étudiants de l'École ainsi qu'un « Who's Who » qui recense les anciens centraliens occupant des positions importantes dans le secteur informatique. Dès 1967, Centrale-informatique organise des « réunions d'informations pour les profanes. Ces réunions pouvant d'ailleurs être des conférences, dîners-débats, ou des journées d'études »³³. À partir de 1968, le groupe organise régulièrement des visites d'entreprises utilisant l'informatique en se servant du réseau des anciens élèves de l'École. En effet, ces visites, destinées aux centraliens volontaires, prennent toujours la même forme : les élèves visitent le centre de calcul d'une entreprise utilisatrice, ils reçoivent un exposé sur le fonctionnement pratique de l'ordinateur en entreprise, les commentaires sont effectués par d'anciens centraliens. Sans énumérer l'ensemble des visites³⁴ effectuées par le groupe entre 1968 et 1973, il est utile de présenter quelques exemples. Les 10 et 17 octobre 1968, une soixantaine de centraliens participent à la visite du centre de traitement de l'information de la SNCF. Ils sont accueillis par Antoine Vosluisant, adjoint au traitement de l'information à la direction des études

32 L'existence du bulletin *C.I.B.L.E.* est mentionnée par la revue *Arts et Manufactures*, nous n'avons retrouvé aucun numéro de celui-ci.

33 Archives Nationales de Fontainebleau. 19820263 Art 2. Réunion de l'association Centrale-informatique du 20 mars 1967, p. 2. Si le groupement professionnel Centrale-informatique est constitué par la réunion du 8 février 1968, il existait sous la forme d'une association dès 1967.

34 Ces visites sont mentionnées par la revue *Arts et Manufactures*.

générales de la SNCF qui leur présente la méthode suivie par la SNCF dans le domaine du traitement de l'information. Ensuite, ils écoutent une intervention de Louis Noël Joly, issu de la promotion 1961 de l'École centrale et directeur de l'assistance technique d'UNIVAC France, qui porte sur les caractéristiques techniques de l'UNIVAC 1108³⁵. Le 17 avril 1969, le groupe professionnel effectue une visite au centre de calcul de la compagnie Shell. Ce centre est dirigé par Laze, centralien de la promotion 1944. Les exposés sont réalisés par Laze ainsi que par J. Baret, chef du service « Calcul économique » et centralien de la promotion 1961³⁶.

En organisant ces visites, le groupe s'appuie sur l'esprit de corps de l'École en intégrant les nouveaux élèves dans les réseaux préexistants de centraliens. Ainsi, si ces manifestations se présentent comme une initiation des élèves aux aspects pratiques de l'informatique en entreprise, elles ont essentiellement pour fonction de démontrer les possibilités offertes aux centraliens de faire carrière dans le secteur. Dans le même temps, elles permettent aux élèves de nouer des contacts pour travailler éventuellement dans l'informatique. Enfin, elles assurent la visibilité des élèves de l'École centrale dans un marché du travail en expansion. La promotion de l'enseignement de l'infor-

35 Groupements professionnels (1969), « Centrale-informatique. Visite du centre de traitement de l'information de la SNCF », *Arts et Manufactures*, 193.

36 Groupements professionnels (1969), « Visite du centre de traitement de la compagnie Shell », *Arts et Manufactures*, 200, p. 42.

matique effectuée par le groupe professionnel consiste donc en une valorisation des carrières de l'informatique. En 1969, Tilloy et Vernières, des promotions 1951 et 1958 de l'école et membres du groupement, publient un article intitulé « Les carrières de l'informatique ». Celui-ci, commandé par le service de placement de l'Association³⁷, se conclut en insistant sur les possibilités de carrières offertes par l'informatique :

Est-il nécessaire d'ajouter qu'il manque beaucoup de personnel et surtout de personnel compétent en Informatique ? Oui certainement, si cela peut vous inciter à rejoindre une profession où vous aurez toutes les chances de faire une carrière brillante et variée³⁸.

La place de l'informatique dans le *curriculum* de l'École centrale

Un enseignement de gestion

À partir de 1967-1968, les enseignements d'informatique et ceux de gestion croissent parallèlement et sont étroitement associés tant dans la formation commune à tous les élèves d'une promotion, c'est-

à-dire les deux premières années de scolarité de l'école, que dans les enseignements de spécialité donnés aux élèves de troisième année³⁹. À partir de l'année scolaire 1967-1968, l'informatique est enseignée lors des deux premières années de scolarité et fait partie d'un ensemble de cours intitulé « Sciences de l'information et de l'économie ». Par cette position dans le *curriculum*, il s'agit d'initier simultanément les étudiants aux « *mécanismes économiques fondamentaux* » et aux « *premières notions d'Informatique* » afin de parvenir à « *une meilleure assimilation de ces matières généralement étrangères à l'élève des classes de mathématiques spéciales* »⁴⁰. À partir de 1970, les étudiants ne travaillent plus uniquement en Fortran, utilisé pour les applications scientifiques, mais aussi en COBOL, langage traditionnel des informaticiens de gestion⁴¹. Plus précisément, l'enseignement d'informatique de première année est présenté comme un cours d'informatique générale et lors de la deuxième année il faut « *situer le rôle de cette dernière technique dans la gestion de l'entreprise et les problèmes de choix de matériels, de personnels et la réorganisation qu'elle implique* »⁴². Dans ce cadre, les élèves étudient les sciences du

³⁷ L'article ne précise pas s'il s'agit de l'Association des anciens élèves de l'École centrale ou de centrale Informatique. Il est néanmoins probable qu'il s'agisse de l'Association des anciens élèves de l'École centrale, Centrale-informatique n'étant une association que jusqu'en février 1968.

³⁸ Tilloy & Vernières (1969), « Les carrières de l'informatique », *Arts et Manufactures*, 202, p. 23.

³⁹ Anonyme (1969), « Assemblée générale du groupement Centrale-informatique du 20 février 1969 », *Arts et Manufactures*, 197, p. 40.

⁴⁰ Boucheron R. (1970). « L'enseignement à l'École centrale », *Arts et Manufactures*, 207, p. 45.

⁴¹ Baron J.-L. (1970), « L'élève ingénieur devant l'entreprise », *Arts et Manufactures*, 210, p. 20.

⁴²

management, les principes de choix des ordinateurs en fonction des problèmes de l'entreprise, l'organisation des fichiers, les problèmes de personnel induits par l'informatisation et le panorama récent du logiciel⁴³, soit autant de questions qui renvoient aux applications de l'informatique à la gestion. L'association de l'informatique à la gestion et la croissance de ces enseignements dans le cursus de l'école est suffisamment importante pour que le directeur de l'École centrale déclare que « *l'enseignement de gestion a fait son plein de ce que nous pouvions mettre en place sans trop compromettre l'enseignement des techniques qui reste [...] l'essentiel de notre formation* »⁴⁴.

Cette association de l'informatique et de la gestion se retrouve dans les enseignements de spécialité lors de la troisième année du cursus de l'École centrale. Les différentes options de « gestion scientifique » proposées aux étudiants entre 1967 et 1975 permettent d'approfondir les notions d'informatique abordées durant les deux premières années de la formation. Parallèlement, l'enseignement de l'informatique permet de donner une légitimité

scientifique à celui de la gestion comme dans les universités où celle-ci se développe⁴⁵. Ainsi, en 1967, l'École centrale propose une option « Sciences économiques et Gestion des entreprises » qui est suivie par une trentaine d'élèves désirant se spécialiser dans ce domaine. La moitié des leçons et des séances de travaux pratiques sont consacrées à l'informatique et au calcul économique. Les travaux effectués par les étudiants de cette spécialité dans le cadre de leur stage obligatoire dans l'industrie relèvent, pour certains d'entre eux, de l'informatique de gestion. Ainsi, parmi les sujets étudiés, figurent, par exemple, « *le traitement automatique de l'information appliqué à la fiabilité et à l'entretien préventif dans une grande compagnie aérienne* », l'« *étude du transfert sur ordinateur du fichier général des entreprises françaises pour une grande banque* » ou encore l'« *analyse et [la] programmation des devis de travaux publics pour le compte d'un constructeur d'ordinateurs* »⁴⁶. À partir de la rentrée 1968 le fonctionnement de la troisième année est réformé. Les anciennes options sont remplacées par sept départements dont un de mathématiques appliquées et de gestion (MAG)⁴⁷. Si ce département

43 Archives de l'École centrale 7.11.4. Ministère de l'Éducation Nationale, École centrale des Arts et Manufactures, « Programme des Études », 1972.

44 Baron J-L. (1970), « L'élève ingénieur devant l'entreprise », *Arts et Manufactures*, 210, p. 20. L'enseignement de gestion désigne ici les enseignements d'informatique et de gestion proprement dite. En effet, l'article range dans la même catégorie d'enseignements « les mécanismes économiques fondamentaux », la « comptabilité », le « calcul économique », l'« organisation du travail », « le droit de l'entreprise », « les relations humaines » et « l'informatique ».

45 Pour une étude de la légitimation croisée entre la gestion et l'informatique je me permets de renvoyer à ma thèse (Neumann, 2013, pp. 352-446).

46 Anonyme (1967), « Où en est l'enseignement de l'informatique à l'École centrale ? », *Arts et Manufactures*, 180, p. 48.

47 Les autres départements sont ceux de physique et métallurgie, chimie, constructions mécaniques, énergétique, construction civile et électricité. Si l'École centrale n'abandonne pas les « enseignements industriels » au profit de la gestion, on peut constater

ne se limite pas à l'informatique⁴⁸, cette dernière joue un rôle important dans le fonctionnement de celui-ci. Ainsi, selon le directeur de l'École centrale, le département MAG doit « *donner à la formation de gestion un caractère plus scientifique, plus informatique et aller dans le sens d'une bonne utilisation des qualités scientifiques des élèves de notre école* »⁴⁹. L'enseignement du département se définit par l'interdépendance entre les mathématiques appliquées, l'informatique et la gestion. La formation en gestion repose sur la programmation linéaire, la théorie des graphes et la gestion des stocks tandis que celle en informatique porte sur le télétraitement, le temps partagé, la multiprogrammation et le multitraitement⁵⁰. De plus, le département MAG propose une option « *Systèmes informatiques* » qui doit permettre aux élèves-ingénieurs d'acquérir les connaissances théoriques et pratiques les rendant aptes à la conception des systèmes informatiques⁵¹.

que cette dernière fait tout autant partie du socle de connaissances générales des diplômés de l'École centrale que les disciplines industrielles puisqu'elle dispose comme celles-ci d'un département permettant une spécialisation en troisième année.

48 Anonyme (1969), « Assemblée générale du groupement Centrale informatique du 20 février 1969 », *Arts et Manufactures*, 197, p. 40.

49 (1968). « Exposé de Jean-Jacques Baron lors du congrès de l'association le 2 mars 1968 », *Arts et Manufactures*, 186, p. 36 et Anonyme (1969), « Assemblée générale du groupement Centrale-informatique du 20 février 1969 », *Arts et Manufactures*, 197, p. 40.

50 Anonyme. (1969), « Assemblée générale du groupement Centrale informatique du 20 février 1969 », *Arts et Manufactures*, 197, p. 41.

51 Teper M. (1970), « Orientation du département

Assimiler la logique informatique pour former des utilisateurs de haut niveau

En considérant l'informatique de gestion comme une matière d'enseignement général et comme un élément de polyvalence professionnelle, l'École centrale a pour but de former des utilisateurs, c'est-à-dire des cadres utilisant l'informatique dans leur travail. Par exemple Baron fixe comme objectif à l'École centrale d'« *être à l'avant-garde de l'utilisation de l'ordinateur* »⁵². Dans une interview accordée à la revue *Euroscopie*, il fait de l'informatique une des disciplines qui doit permettre aux centraliens de prendre la direction d'un nombre de projets variés. Ainsi les « *enseignements de gestion, de mathématiques appliquées, d'informatique appliquée* » doivent permettre :

De former des ingénieurs généralistes qui seront à même de prendre un projet d'ensemble comme une fusée, un réacteur nucléaire, un pétrolier de 500 000 tonnes et qui, ayant une formation dans un grand nombre de disciplines seront capables de diriger ces complexes que seront les réalisations du monde moderne⁵³.

En ayant pour objectif de former des utilisateurs de haut niveau, l'École centrale doit donner un enseignement

mathématiques appliquées et gestion », *Arts et Manufactures*, 207, p. 82.

52 Anonyme (1969), « Assemblée Générale du groupement Centrale informatique », *Arts et Manufactures*, 197, p. 41.

53 Anonyme (1971), « Interview de J.-J. Baron », *Euroscopie*, 12, 1971, p. 18.

suffisamment poussé pour permettre le dialogue avec les informaticiens professionnels tout en étant destiné à des étudiants non-informaticiens. Il s'agit donc de limiter l'étendue des thèmes abordés afin de se distinguer d'une école spécialisée dans l'informatique tout en transmettant une connaissance suffisamment fondamentale de l'informatique pour permettre aux futurs diplômés de pouvoir réactualiser leurs connaissances techniques au cours de leurs carrières.

L'acquisition de cette connaissance fondamentale passe par l'apprentissage de la programmation. Celle-ci constitue l'ossature des cours de première année d'informatique et « l'étape initiale » d'un apprentissage ultérieur⁵⁴. À la fin du 1^{er} trimestre les élèves de centrale doivent connaître le langage Fortran puis à partir de 1970 sont initiés au Cobol pour une durée de 38 heures⁵⁵.

Le choix de la programmation peut s'expliquer d'une part, parce que celle-ci est un des objets de l'informatique

théorique⁵⁶, ce qui la rend à même de constituer le socle fondamental désiré par l'École centrale, d'autre part, parce que l'informatisation d'une entreprise est impossible sans la confection de programmes. La programmation se situe donc à la fois dans le domaine fondamental de l'informatique et dans celui de ses applications, comme en témoigne la multiplication des méthodes de programmation destinées aux entreprises à la fin des années 1960. Dans le cursus de centrale, l'apprentissage des autres aspects de l'informatique vient après la programmation qui est assimilée à une logique spécifiquement informatique. Ainsi, le positionnement de l'enseignement de la programmation avant celui de l'analyse⁵⁷ est présenté comme une solution contre-intuitive mais plus formatrice :

On peut se demander pourquoi le cours d'analyse vient après le cours de programmation, la logique était d'inverser mais finalement on ne constate pas un mauvais résultat puisque les élèves à partir du moment où ils connaissent la manipulation du Fortran sont désireux de trouver des domaines d'application et de mieux comprendre l'utilité de leur outil et finalement le résultat de cette expérience est assez favorable⁵⁸.

⁵⁴ Ce point est précisé dans les sources suivantes : Anonyme (1969), « Assemblée Générale du groupement centrale informatique », *Arts et Manufactures*, 197, Boucheron R. (1970), « L'enseignement à l'École centrale », *Arts et Manufactures*, 207, et Archives de l'École centrale. 7.11.4. Ministère de l'éducation Nationale, École centrale des Arts et manufacture, « Programme des études » 1972.

⁵⁵ Archives nationales de Fontainebleau, 19810642 Art 14. SERTI (Société d'études et de réalisations pour le traitement de l'information), *Enquête sur la formation à l'informatique dans les grandes écoles d'ingénieurs*, le 6 mars 1970 pp. 13-14 et Baron J.-J. (1970), « L'élève ingénieur devant l'entreprise », *Arts et Manufactures*, 210, p. 20.

⁵⁶ Elle peut se concevoir indépendamment d'une application ou d'un matériel informatique.

⁵⁷ Il s'agit de l'analyse fonctionnelle c'est-à-dire l'ensemble des études préalables définissant les problèmes de gestion et le schéma de circulation de l'information. Celle-ci détermine l'analyse organique c'est-à-dire la solution machine.

⁵⁸ Anonyme (1969), « Assemblée Générale du groupement Centrale informatique », *Arts et*

Parallèlement, l'École désire promouvoir un « enseignement intégré » de l'informatique. L'enseignement de gestion se confond en partie avec celui de l'informatique et des mathématiques appliquées tant dans le tronc commun de la formation qu'au niveau du département MAG pour la troisième année d'études. Cependant, si l'enseignement de l'informatique relève de la gestion, l'ordinateur doit pouvoir être utilisé comme outil dans toutes les disciplines présentes à l'École centrale. L'utilisation généralisée de l'informatique s'inscrit dans le projet de renforcer la polyvalence des centraux en supprimant les compartiments entre les différentes disciplines. Ainsi, R. Boucheron, le directeur adjoint de l'École, peut écrire que la polyvalence recherchée par l'école « *n'est pas celle résultant de l'acquisition d'une somme de nombreux compartiments renfermant méthodes expériences et recettes* » mais « *celle d'une multiplicité de liaisons offertes [...] aux éléments scientifiques et techniques en création et en variation incessante* »⁵⁹. L'informatique contribuerait donc à éviter que la formation de l'École centrale ressemble à un catalogue de disciplines superficiellement étudiées et sans rapport les unes avec les autres en constituant un outil commun à celles-ci. Cet objectif est rappelé explicitement lors de l'Assemblée générale du Groupe Central-informatique en 1969 :

Manufactures, 197, p. 41.

⁵⁹ Boucheron R. (1970), « L'enseignement à l'École centrale », *Arts et Manufactures*, 207, p. 46.

En troisième année chacune des spécialités va maintenant utiliser l'informatique pour ses propres enseignements et ses propres projets ; autrement dit les deux premières années ont pour mission d'assurer l'enseignement de l'informatique comme outil nouveau de recherche, et de calcul, et en troisième année nous procédons à l'application de cet outil dans les différentes spécialités⁶⁰.

L'utilisation de l'informatique par tous les départements de l'École ne remet pas en cause l'assimilation de celle-ci à la gestion. En réalité, pour la direction de l'École centrale, il s'agit d'établir des passerelles entre les nouveaux enseignements de gestion et ceux plus anciens et plus légitimes de sciences industrielles car l'utilisation de l'informatique, par ces derniers, repose sur l'activité du département MAG. En effet, celui-ci « *doit former des spécialistes qui apporteront leur concours aux autres départements pour le développement de l'informatique aux différentes sciences et techniques enseignées à l'école* »⁶¹. De même, il est qualifié de « *noyau de la pluridisciplinarité* » de l'École car il doit fournir des assistants en informatique aux autres départements et rendre possible la constitution « *entre ses propres élèves ingénieurs et ceux des autres départements, [d'] équipes pluri-*

⁶⁰ Anonyme (1969), « Assemblée Générale du groupement Centrale informatique », *Arts et Manufactures*, 197, p. 40.

⁶¹ Baron J.-J. (1970), « La nouvelle école : ce qu'il reste à faire », *Arts et Manufactures*, 207, p. 31.

disciplinaires, notamment pour l'étude en commun de certains projets »⁶².

La pédagogie informatique entre contraintes pratiques et normes managériales

À l'École centrale, l'enseignement de l'informatique repose sur un accès facile des élèves à l'ordinateur ou au centre de calcul : « *Les élèves disposent librement des consoles dans la limite de possibilités pour des exercices personnels ou des résolutions de problèmes relatifs aux autres disciplines* »⁶³. Cette situation apparaît d'abord comme une conséquence du projet d'intégration de l'informatique aux autres disciplines tout en permettant de résoudre certaines difficultés pratiques lors du démarrage de l'enseignement. En effet, elle permet la constitution d'une bibliothèque de programmes prête à être utilisée dans une gamme d'enseignements variés⁶⁴.

Ce programme pédagogique doit être analysé comme un apprentissage de l'ethos professionnel des cadres dirigeants qui se met en place dans les années

soixante. Celui-ci insiste sur l'autonomie et la créativité des cadres dans des entreprises dont les différents secteurs doivent eux-mêmes être décloisonnés. Cette préoccupation, commune à l'ensemble des écoles d'ingénieurs et de commerce, est cependant plus prégnante dans les écoles généralistes, comme l'École centrale, d'une part, parce qu'elles se donnent pour vocation de former les cadres de direction, d'autre part, parce que l'informatique permet à cette ambition de s'incarner dans des supports pédagogiques concrets, tels les jeux d'entreprises, en mettant les capacités de calcul des ordinateurs au service des différents enseignements.

À l'École centrale, l'introduction de l'enseignement de l'informatique s'inscrit ainsi dans un contexte de promotion des « méthodes actives » censées développer chez les élèves l'autonomie et le sens des responsabilités qui seraient le propre des cadres dirigeants. À Centrale, le projet d'enseignement non directif passe par la mise en place d'un régime de scolarité qualifié de « libéral » qui doit permettre d'éviter de « *transformer l'activité créatrice et le dynamisme propre des jeunes gens en une attitude passive de récepteur restituant le plus fidèlement possible ce qu'il leur a été dit ou ce qu'ils ont appris.* »⁶⁵ Ainsi, à partir de la fin des années 1960, seule la présence aux travaux pratiques est obligatoire à l'École⁶⁶. Dans le discours qu'il

⁶² Teper M. (1970), « Orientation du département mathématiques appliquées et gestion », *Arts et Manufactures*, 207, pp. 81-82.

⁶³ Archives nationales de Fontainebleau, 19810642 Art 14. SERTI (Société d'études et de réalisations pour le traitement de l'information), « Enquête sur la formation à l'informatique dans les grandes écoles d'ingénieurs », le 6 mars 1970, p. 13.

⁶⁴ Bratière J. & Jeuland A. (1971), « Pour une conception nouvelle du département Math-Appli-Informatique », *Piston*, 1, p. 14.

⁶⁵ Baron J.-J. (1969), « Motivation-Créativité », *Travail et Méthodes*, 6, p. 20.

⁶⁶ Anonyme. (1969), « Des élèves de Centrale ont la

prononce lors de l'assemblée générale du Groupement Centrale-informatique, le Directeur de l'École fait de l'informatique la discipline qui permettrait au régime « libéral » de l'École de déboucher réellement sur un travail autonome de la part des étudiants :

Mais je voudrais aussi faire appel à l'informatique pour résoudre les problèmes qui sont des problèmes de base dans la nouvelle orientation de notre école. Vous savez que j'ai pris le risque de supprimer, pratiquement, toutes les contraintes à l'École, c'est-à-dire qu'il n'y a plus de points de discipline, il n'y a plus de points d'assiduité, ce régime qui assurait le remplissage des amphithéâtres, pas le remplissage intellectuel mais le remplissage physique des amphithéâtres. C'était très réconfortant pour un directeur d'École, il n'avait qu'à voir que tous les gradins étaient remplis, mais ce n'était qu'une illusion⁶⁷.

À l'École centrale, l'enseignement de l'informatique renvoie donc à l'importance accrue de la gestion – conçue non plus comme un ensemble de propriétés personnelles des dirigeants mais comme une discipline qui doit être formalisée et enseignée – dans la définition du profil des cadres dirigeants. Cependant, la gestion et l'informatique demeurent des disciplines illégitimes pour une partie des étudiants.

parole », *Travail et Méthodes*, 6, 1969, p. 23.

⁶⁷ Anonyme. (1969), « Assemblée Générale du groupement Centrale informatique », *Arts et Manufactures*, 197, p. 41.

Résister à la conversion managériale de l'École centrale

La contestation des enseignements de gestion

Si les professeurs et les dirigeants de l'École centrale se félicitent du succès rencontré par la spécialisation en informatique effectuée en troisième année, les réactions des étudiants ne sont pas unanimement favorables. L'hostilité d'une fraction des étudiants à la nouvelle discipline s'inscrit dans un mécontentement plus large vis-à-vis du renforcement du rôle de la gestion et du management dans le cursus de l'école. Par exemple, en 1972, une vingtaine de copies blanches sont rendues à l'issue d'un cours de gestion de Teper en troisième année⁶⁸. L'année suivante, 120 élèves de troisième année refusent de composer à un examen de gestion⁶⁹. Par rapport aux autres disciplines enseignées à l'école, la

⁶⁸ *Piston*, 9.3, 1972. *Piston* est la revue du Bureau des étudiants (BDE) de l'École centrale. Elle est conservée aux archives de l'École. Dans les archives, les années 1972-1973 apparaissent comme celles de la contestation la plus brutale de l'enseignement de gestion. Cela ne signifie aucunement que la critique des différentes disciplines de gestion se limite à ces années-là. Il s'agit plutôt de l'aboutissement d'un processus, débuté en 1968, dans lequel la direction de l'École s'oppose régulièrement au BDE qu'elle estime trop politisé. L'année 1973 permet de voir avec le plus de clarté la logique des arguments des étudiants opposés au développement des cours de gestion sur la période 1968-1975.

⁶⁹ Baron J.-J. (1973), « Exposé de Baron lors du congrès de l'Association des anciens élèves de l'École centrale le 31 mars 1973 », *Arts et Manufactures*, 241, p. VII.

gestion souffre d'une légitimité scientifique moindre. Le caractère superficiel de son enseignement est rapporté, soit aux limites intrinsèques de la discipline, soit à l'absence de qualité pédagogique des enseignants. Ainsi Baron remarque que le refus de composer lors de l'examen de gestion de 1973 s'explique par l'attachement des étudiants à une conception technique et plus rigoureusement scientifique du rôle de l'ingénieur :

Ces 120 élèves contestataires, je les ai eus en face de moi à l'amphi pendant deux heures. J'ai voulu m'expliquer. Dans les arguments qui m'ont été donnés, celui qui est revenu le plus souvent, c'est celui-ci :

« Nous sommes à l'École centrale, nous comptons faire de la recherche, des sciences. Nous n'avons pas besoin d'enseignement de gestion, d'enseignement d'économie. Nous voulons seulement être des techniciens purs, dans tous les sens du terme. »⁷⁰

Une enquête de la plus importante SSII française, la Société d'économie et de mathématiques appliquées (SEMA), commanditée par l'association des anciens élèves de l'École centrale en 1973, constate qu'une « *petite minorité* » des élèves manifeste un refus absolu des cours de gestion :

D'une certaine manière, les élèves refusent d'entrer dans la course aux responsabilités et au pouvoir, comme ces gestionnaires habiles à se mettre

en valeur à travers l'exercice d'un vocabulaire dissimulant leur manque de profondeur et de compétence. Ce qui compte, pour les élèves, c'est d'avoir un métier véritablement intéressant, c'est-à-dire dans la production, domaine de la création, et de la compétence véritable⁷¹.

Parallèlement, les élèves favorables aux cours de gestion déplorent la qualité médiocre de ceux-ci :

La plupart dénoncent le caractère très superficiel de l'enseignement qu'ils reçoivent en cette matière. Regrettant peut-être que la gestion ne soit pas une science exacte, ils ont l'impression qu'on leur offre plus un vernis destiné à leur permettre d'avoir l'air au courant qu'une véritable formation à cette discipline⁷².

L'informatique, une discipline illégitime

On retrouve des arguments identiques dans les critiques formulées à l'égard des cours d'informatique. En juin 1970, l'Union nationale des étudiants de France-Union des grandes écoles (UNEF-UGE) de l'École centrale publie un long réquisitoire contre le cours d'informatique de J.-P. Bouhot dans *Piston*, rebaptisé, pour l'occasion, *Piston-Intoxication*. Le cours de Bouhot est jugé inconsistant par les auteurs de l'article :

⁷¹ Archives de l'École centrale. 9.1.1. SEMA, Association des anciens élèves de centrale, *Image de l'école centrale*, 1973, p. 167.

⁷² *Id.*, p. 169.

⁷⁰ *Id.*, p. VII-VIII.

Deux choses caractérisent ce cours :

- Le contenu : ce n'est qu'un insipide baratin, au mieux de la mauvaise vulgarisation en tout cas ne justifiant jamais l'existence de ce cours.
- Les digressions philosophiques : toutes les trois minutes, le prof éprouve le besoin d'ouvrir une parenthèse, soit pour nous faire part de son expérience « personnelle », soit pour restituer ce qu'il vient de dire dans un contexte toujours plus ou moins métaphysique, afin de nous faire bien voir à quelle époque nous vivons et dans quelle société nous aurons à travailler. Le malheur est que cette conversation de salon, qu'il tente le brave homme, d'établir entre l'amphi et lui, aurait plutôt l'allure d'un bourrage de crâne intense⁷³.

Pour l'UNEF, le cours ne repose pas sur des résultats et des exercices concrets mais sur l'accumulation d'affirmations gratuites portant sur les conséquences sociales de l'informatique : « *Méfiance d'abord, quand Bouhot étale son expérience personnelle, car tous ses grands slogans, ses affirmations péremptaires se retrouvent texto dans n'importe quel bouquin d'informatique. Il y a donc peu de chances pour que tout vienne de la pratique et des problèmes effectivement rencontrés* »⁷⁴. Outre l'aspect stéréo-

typé des cours proposés, l'UNEF remet en cause la valeur disciplinaire et scientifique de l'informatique. Celle-ci est réduite à une recherche de la rentabilité dépourvue de valeur scientifique. Dans ce raisonnement, travailler dans l'informatique revient à sous-employer les capacités scientifiques des centraliens :

Quand Bouhot nous présente l'informatique comme la voie royale, on peut être sceptique pour plusieurs raisons : outre le pitoyable spectacle qu'il offre personnellement, il faut noter son carriérisme forcené (on ne peut causer avec lui plus de 5 minutes sans qu'il parle de « *la magnifique carrière qu'on peut faire dans l'informatique* »), ce qui doit contribuer à obscurcir sa pensée à ce sujet. Plus précisément, constatons qu'en France, l'informatique c'est presque exclusivement l'informatique de gestion. Quand on connaît la prolifération souvent parasitaire des secteurs de « gestion » au sens large dans l'économie, les sinécures ne doivent pas manquer. Quant à l'autre aspect, le contrôle de processus industriel [...] ce n'est pas toujours très beau du point de vue scientifique et encore une fois ça ne peut signifier qu'une rentabilisation du dit processus. Notons enfin que si beaucoup de pistons rentrent à IBM, c'est pour faire du technico-commercial, et qu'il n'y a pas de différences entre un centralien technico-commercial ou commercial chez IBM et un camelot bien formé, bien payé et muni d'un diplôme⁷⁵.

⁷³ UNEF-UGE, Anonyme, « L'informatique dans la société », *Piston-Intoxication*, 18 juin 1970, 1.I (non paginé).

⁷⁴ *Id.* 2.I. De la même manière l'auteur note aussi : « Puis, tout au long du cours, l'informatique est systématiquement décrite comme un domaine à part, éblouissant et très complexe. Par exemple "la conception d'un système d'exploitation, c'est la tâche

la plus complexe qu'ait jamais accompli l'homme", bel exemple d'affirmation gratuite. » *Id.* 1.II.

⁷⁵ *Id.* 2.III.

L'article conclut à l'absence de légitimité de ce cours au sein de l'École centrale :

Pour conclure, on ne peut que constater 3 choses à propos de ce cours

- _absence de contenu justifiant un examen.
- _coloration idéologique confinant au bourrage de crâne.
- _résultat immédiat de la mainmise du patronat⁷⁶.

Dans le cas de l'UNEF, l'opposition au cours de Bouhot a un caractère politique nettement marqué. Cependant, elle semble avoir eu des échos au-delà du syndicat. En effet, dès 1971, la direction de l'école réforme l'organisation de ce cours en réduisant la place des cours magistraux, sur lesquels portait l'essentiel des critiques formulées par l'UNEF : « *Le cours d'informatique de M. Bouhot en 1^{re} année et en 2^e année sera transformé pour comporter désormais moins de séances à l'amphithéâtre et plus de travaux pratiques* »⁷⁷. Dans la nouvelle organisation du cours la place des TD et des travaux pratiques est renforcée avec des séances de TD effectuées en demi-promotion. Malgré cette réforme, le cours continue de susciter l'hostilité ou l'incompréhension d'une partie des étudiants. En 1972, le BDE distribue un questionnaire afin de recueillir les réactions des centraliens sur la qualité de l'enseignement en informatique. Les étudiants critiquent toujours

les cours en amphi, qui sont « *vagues et inexacts* », tout en insistant sur les défauts pédagogiques du cours magistral :

Une grosse majorité a insisté sur les défauts de ce cours, qui sont essentiellement la confusion, l'éloignement du polycopié, le nombre trop important de sujets abordés et par conséquent, vu le manque de temps, la façon superficielle de les traiter. Nombreux sont ceux qui ont reproché le manque d'articulation du plan (certains ont même signalé qu'ils n'en avaient pas vu), le peu de clarté des objectifs du cours, et surtout, ils ont le sentiment que l'exposé n'était pas assez progressif, ne définissant pas les termes nouveaux avec précision, bref ne s'adressait pas à un auditoire complètement neuf en informatique⁷⁸.

Dans le même temps, les critiques s'étendent aux TD. En effet, si « *la compétence [des assistants] n'est pas mise en doute [...] [leurs] qualités pédagogiques et [leur] conscience professionnelle sont très inégales* »⁷⁹.

Conclusion

Alors qu'en 1965 l'enseignement de l'informatique est inexistant dans la quasi-totalité des grandes écoles, en 1975, la discipline est présente dans le cursus de la majorité des écoles d'ingénieurs et de commerce. Cette généralisation correspond à une volonté de moderniser

⁷⁶ *Id.* 2.III (sic).

⁷⁷ *Piston*, 10 mai 1971, p. 4

⁷⁸ *Piston*, 1972, p. 20.

⁷⁹ *Id.*, p. 20.

la formation de ces établissements en s'inspirant des pratiques de formation américaines et en s'ajustant à l'évolution présumée des techniques employées dans les entreprises. En retour, l'incorporation de l'informatique au cursus des grandes écoles produit des effets sur le profil des informaticiens, initialement formés essentiellement par les constructeurs, et des utilisateurs dans les entreprises. D'une part, elle donne consistance et réalité à l'équivalence établie entre les cadres et l'informatique, d'autre part, elle participe à l'élévation du capital culturel des salariés maîtrisant l'informatique, qu'il s'agisse d'informaticiens ou d'utilisateurs occasionnels (Neumann, 2013, pp. 245-253). Cette promotion de l'informatique comme outil propre aux cadres n'est pas sans ambiguïtés car l'informatique n'est jamais enseignée pour elle-même dans les écoles d'ingénieurs mais toujours comme instrument d'une autre discipline (*ibid.* ; Neumann, 2020). Ainsi, à l'École centrale, l'enseignement de l'informatique ne prend sens que rapporté à l'objectif plus général de former ses étudiants au management.

Bibliographie

Boltanski L. (1982). *Les Cadres, La formation d'un groupe social*. Paris : Minuit.

Boltanski L. & Chiapello E. (1999). *Le nouvel esprit du capitalisme*. Paris : Gallimard.

Bourdieu P. (1989). *La Noblesse d'État*. Paris : Minuit.

Chessel A.-M. & Pavis F. (2001). *Le technocrate, le patron et le professeur. Une histoire de l'enseignement supérieur de gestion*. Paris : Belin.

Fourquin J.-C. (2008). *Sociologie du curriculum*. Rennes : PUR.

Harris A. & Sédouy A. de (1977). *Les patrons*. Paris : Seuil.

Lazuech G. (1998). « Le processus d'internationalisation des grandes écoles françaises ». *Actes de la recherche en sciences sociales*, 121-122, pp. 66-76.

Le Roux R. (2018). *Une histoire de la cybernétique en France (1948-1975)*. Paris : Classiques Garnier.

Mounier-Kuhn P. (2010). *L'informatique en France. De la seconde guerre mondiale au Plan Calcul, l'émergence d'une science*. Paris : PUPS.

Neumann C. (2008). « La construction sociale d'une catégorie : les automatismes de gestion ». *Entreprises & Histoire*, 52, pp. 85-95.

Neumann C. (2013). « De la Mécanographie à l'informatique. Les relations entre catégorisation des techniques, groupes professionnels et transformation des savoirs managériaux ». Thèse d'histoire contemporaine, Université Paris-Ouest Nanterre La Défense, 2013.

Neumann C. (2020). « L'immatériel de

direction ». In V. Boussard, M.-A. Dujarier & F. Ricciardi (dir.). *Les Travailleurs du management. Acteurs, dispositifs et politique d'encadrement*. Toulouse : Octarès, pp. 43-65.

Neumann C. (2022). « La mise en récit managériale des ordinateurs en France dans les années 1960 ». *Artefact*, 16, pp. 111-133.

Pavis F. (2003). « L'institutionnalisation universitaire de l'enseignement de gestion (1965-1975) ». *Formation-Emploi*, 83, pp. 51-63.

Tricot M. (2008). *Le moment cybernétique : La constitution de la notion d'information*. Paris : Champ Vallon.

L'utilisation des tests de sélection des informaticiens des années 1960 aux années 1990 : une hiérarchisation des formations et des publics

Michaël Vicente

COSTECH, Université de Technologie de Compiègne

Résumé

Cet article interroge l'introduction et la diffusion des tests de sélection des informaticiens, d'abord aux États-Unis puis en France, depuis les années 1950 jusqu'à la fin des années 1990. En nous concentrant sur le temps long, nous verrons qu'en fonction du type d'acteur il est fait un usage différencié de ces tests et que leur utilisation (et leur non-utilisation) relève de fonctions sociales qui participent à une hiérarchisation des formations et des publics à qui ils s'adressent.

Mots-clés : informatique ; recrutement ; formation ; tests de sélection, formation privée.

Abstract

This article examines the introduction and diffusion of selection tests for computer programmers, first in the United States and then in France, from the 1950s to the end of the 1990s. By focusing on the long term, we will see that, depending on the type of actor, these tests are used differently and that their use (and non-use) is a function of social functions that contribute to a hierarchy of training and of the public for which these tests are intended.

Keywords: IT; recruitment; training; selection tests; private training.

Introduction

L'usage des tests d'aptitude pour l'orientation ou le recrutement dans les professions techniques possède une longue histoire (Huteau, 1996). Comme l'a étudié Cédric Neumann (2013), en France, dans la continuité de leur utilisation pour la mécanographie, les tests ont été utilisés lors de l'introduction des ordinateurs dans la deuxième moitié des années 1950 pour recruter les programmeurs, au sein de cette profession naissante. Dans cet article, il s'agit dans un premier temps de voir comment des tests de sélection standards ont été développés aux États-Unis pour répondre à une demande d'informaticiens et de comprendre pourquoi, à ce moment précis, leur utilisation est largement promue dans le monde de la formation. En nous intéressant plus précisément à la manière dont ils ont été introduits dans les organisations, nous verrons que ces derniers ont suscité de nombreuses controverses. Ensuite, nous verrons que s'ils se maintiennent en dépit des controverses, leur utilisation est fort différenciée. Une différenciation que l'on observe entre la France et les États-Unis d'une part, et entre la sélection pour la formation ou pour le recrutement par les entreprises d'autre part, mais aussi au sein même du monde de la formation. En analysant son importation en France, nous verrons notamment comment, dans les années 1970, un usage opportuniste et commercial des tests dans les formations privées a pu être fait. Nous analyserons cette différenciation comme la marque d'une

hiérarchisation des publics et des types de formations. En effet, en interrogeant l'utilisation moindre de ces tests, que l'on peut observer à partir des années 1980, nous verrons s'il s'agit d'une spécificité de l'informatique ou s'il s'agit du fait que ces tests se retrouvent en concurrence avec d'autres types de dispositifs de sélection plus généralement utilisés.

Nous adopterons une démarche socio-historique visant à rendre compte des conditions socio-économiques et scientifiques ayant amené à l'apparition de ces tests pour le recrutement. À cet effet, nous avons analysé la littérature scientifique et managériale (principalement américaine) propre à la conception des tests. On retrouve cette littérature produite par des universitaires, mais aussi des chercheurs travaillant dans les grandes entreprises du secteur, non seulement dans des revues comme celles de l'Association of Computer Machinery (ACM) mais également dans des revues propres à la psychologie du recrutement du type *Personnel Psychology*. Dans un deuxième temps, en nous intéressant au cas français, nous nous interrogeons sur l'usage concret de ces tests, en analysant systématiquement la presse professionnelle de l'époque. Il faut en effet voir dans cette presse un moyen de promotion de l'informatique, mais aussi et surtout comme le rappelle Vettraino-Soulard (1983), une source d'information permanente pour les utilisateurs et les acteurs de l'informatique. Si, en France, à partir du milieu des années 1970, de nombreuses revues apparaissent, comme *L'ordinateur*

individuel, *Décision informatique*, ou encore *Informatique et gestion* qui se présentent comme étant destinées au grand public, avec notamment un rôle de vulgarisation, nous avons ici choisi de nous intéresser plus particulièrement à la revue *Ol.informatique*. En effet, créée en 1966, cette revue couvre non seulement entièrement la période étudiée, mais traite également des préoccupations des recruteurs et personnes chargées de la sélection des informaticiens. Nous avons ainsi pu systématiser l'analyse des articles et petites annonces, que nous avons complété avec l'analyse de la presse généraliste : *Le Monde* et le *Nouvel Observateur*.

Les tests de sélection des informaticiens : naissance et controverses

A priori signe d'objectivité, on pourrait supposer que ces tests font l'unanimité. Or, en nous appuyant sur l'étude de la littérature de l'époque, on peut voir que leur introduction a été problématique, tant dans l'efficacité intrinsèque de ces tests que sur plan contextuel plus général.

Des tests de sélection standard développés aux États-Unis pour répondre à une demande d'informaticiens

Aujourd'hui encore, aux États-Unis les tests de sélection dans le système éducatif américain sont très présents. À la fin

des études secondaires, le *Scholastic Aptitude Test* (SAT) est ainsi le passage obligé pour intégrer l'enseignement supérieur. Sur le modèle du test de Binet¹, on introduit ces tests au début des années 1930 dans le système d'enseignement américain. En suivant Nicholas Lemann (1999), on peut voir que s'ils ont une vocation généraliste, dès le début des liens forts entre les tests généralistes en développement et l'industrie mécanographique de l'époque (notamment dans la firme IBM) peuvent être cependant observés. C'est ainsi que durant la grande dépression, et à partir du constat de la faillite des élites, Henry Chauncey, qui fondera l'ETS (*Educational Testing Service*), porte, alors qu'il est assistant à l'Université de Harvard, l'idée d'une démocratisation visant à accueillir davantage d'étudiants en les sélectionnant différemment et de manière plus homogène. Sous l'impulsion de la fondation Carnegie, la *Progressive Education Association* réfléchit à d'autres manières de recruter dans l'enseignement pour ceux qui ne sont pas titulaires de diplômes. Le projet nommé Pennsylvania Study est porté par Ben Wood de l'université de Columbia, le chercheur en éducation William Learned et Henry Chauncey : « *l'idée était d'établir un ensemble de supports que tous les étudiants du lycée ou de l'université devraient être tenus de maîtriser, de les tester et sélectionner la population étudiante sur la base des ré-*

¹ Le test d'Alfred Binet introduit en 1907 l'« échelle métrique de l'intelligence » selon une approche de psychologie expérimentaliste, initialement conçue dans un but pédagogique visant à détecter les écoliers anormaux (Pinell, 1995).

sultats » (*ibid.*, p. 22-23). C'est à l'issue du projet Pennsylvania study qu'IBM intervient via une collaboration directe entre Thomas Watson (fondateur d'IBM) et Ben Wood. Ce dernier avait auparavant créé la Cooperative Test Service qui vendait des tests pour les écoles élémentaires et universités. Le but de cette collaboration avec IBM était alors de massifier et d'industrialiser ce processus et de pouvoir évaluer des centaines de milliers, voire des millions de personnes (*ibid.*, p. 35). Alors qu'au sein d'IBM la technologie n'a pas pu être mise au point en interne, c'est grâce au rachat d'une innovation développée par un enseignant de science Reynold Johnson (*ibid.*, p. 37), le « Markograph » qu'ils ont pu, dès 1936, réaliser une version propre à IBM, et qui sera finalement utilisée par les lycées de Rhode Island et de Providence (*ibid.*, p. 38). Ce n'est certes pas la technologie qui sera adoptée par le fameux test SAT, néanmoins la participation de la firme informatique est indéniable dans le développement des tests dans l'enseignement secondaire et universitaire aux États-Unis².

Pour la construction des grands projets, l'histoire des tests débute réellement en 1952, autour du projet SAGE, *Semi-Automatic Ground Environment*. Il s'agit d'un projet militaire de détection aérienne, qui sera le premier déve-

loppement logiciel d'ampleur à échelle industrielle. Alors que le système éducatif en informatique est encore balbutiant, le développement d'un tel projet nécessite de recruter en nombre des individus, les former rapidement, afin qu'ils deviennent des programmeurs. Hasan Ozbekhan, qui a été de 1951 à 1956 directeur de la planification d'une société spécialement créée pour le programme SAGE, relate cette période :

Nous avons établi des critères pour définir ces spécialistes. [...] Nous sommes descendus dans la rue, nous avons attrapé les passants, nous sommes allés dans les universités chercher les cancrenards qui ne voulaient pas passer leurs examens. Et même dans les prisons ! Puis nous avons mis au point en toute hâte une méthode d'enseignement qui négligeait tous les diplômés. Et nous avons « sorti » un programmeur valable en six semaines. Il fallait inventer. (Cité par Garric, 1972, p 133.)

Il s'agit donc de recruter en masse. Les diplômés n'existant pas, l'idée est de détecter les aptitudes par des tests et de prévoir les capacités à occuper un poste de programmeur. Historiquement, c'est la société SDC (*System Development Corporation*), émanation du RAND (Research ANd Development), qui prendra en charge le développement informatique du projet et mettra en place un management des équipes reposant sur une organisation proche de ce qui était pratiqué dans l'industrie, à savoir une division scientifique du travail : tâches répliquables, simples et répétitives (Ens-

2 Par la suite, le Chauncey group, filiale d'ETS, développera également des formations pour l'aviation et l'informatique, qui en revanche ne deviendront pas des standards.

menger. 2010, p. 61). En cela, ils suivent les principes du taylorisme industriel, où il s'agit de sélectionner scientifiquement le personnel, sur la base de critères objectifs. En testant les aptitudes des programmeurs, telles que la capacité à penser logiquement et à faire des raisonnements abstraits, on donne une « validité prédictive » à leur recrutement, sur leur capacité à occuper un poste.

... qui se maintiennent en dépit des controverses

Le développement de ces tests n'a rien d'évident et suscite des controverses. Nous en relevons trois principales : la première porte sur la méthode à adopter pour construire les tests, ces choix n'étant pas uniquement techniques, la seconde porte sur l'efficacité réelle de ces tests et enfin la dernière est centrée sur la pertinence de leur utilisation en entreprise.

Se pose en effet la question de savoir quel type de méthode adopter. En 1958, Thomas Rowan ingénieur chargé de mettre en place ces tests pour la société SDC, oppose ainsi deux méthodes : la méthode « expert » (*Armchair*) et la méthode de « suivi » (*Follow up*). Dans la première, on détermine *a priori* les qualités jugées nécessaires pour réussir dans le travail considéré. Des tests sont ensuite conçus ou on adopte des tests existants censés mesurer ces qualités. Cette méthode est néanmoins considérée comme hasardeuse, car on ne peut pas « tester » *a posteriori* la validité de

la procédure³. La méthode de « suivi », quant à elle, repose exclusivement sur une analyse précautionneuse du travail, qui nécessite une analyse détaillée de la tâche et qui suppose que celle-ci soit préalablement définie. Cette analyse doit déboucher sur une liste d'aptitudes requises pour ces tâches, et par la suite mettre en place des tests pour évaluer chacune de ces aptitudes (Rowan, 1958, p. 349). C'est cette deuxième méthode que l'on entend appliquer pour les tests d'aptitudes, ainsi que dans le recrutement, et qui sera largement utilisée dans le cadre de ce projet. Dans un premier temps, ce test « maison », assez spécifique, car lié au projet, n'a pas dépassé les frontières de cette entreprise. Ce n'est pas le cas du test initié par la firme IBM, acteur majeur de la production de matériel qui en développant le PAT (*Programming Aptitude test*), propose un test qu'on appellera de seconde génération et qui deviendra le test étalon pour le recrutement de nombreuses entreprises, jusqu'à devenir le passage obligé pour entrer dans la profession⁴ (Ensmenger, 2010, p. 65).

C'est principalement Walter McNamara, psychologue employé chez IBM, qui participera à l'élaboration de ces tests.

³ Thomas Rowan, prend ainsi l'exemple des « Boucleurs de bonneterie ». Ces personnes font un travail très détaillé sur des bas ou chaussettes. On pourrait supposer que cette activité nécessiterait une grande acuité visuelle, et donc dans un premier temps ont été sélectionnées des personnes qui avaient une bonne vue. Or des études ultérieures ont montré de manière contre-intuitive que les personnes myopes étaient en réalité plus efficaces, car plus concentrés (Rowan, 1958, p. 349).

⁴ Ce test est encore utilisé sous le nom « *IPAT test* ».

Lui et son équipe font le constat qu'un grand nombre de tests psychologiques existent déjà sur le marché⁵. Cependant, comme le font remarquer McNamara et Hugues (1961), si ces tests sont utilisés pour de nombreuses professions, aucun d'entre eux ne concerne directement la programmation. Il s'agit donc d'en concevoir qui soit spécifiquement adapté à la profession de programmeur, en prenant en compte ses spécificités. Pour cela, ils s'appuient sur les quelques expertises présentes dans la littérature de l'époque sur les informaticiens. Ils retiennent de Richard Canning (1957), de l'université de Californie à Los Angeles, qu'un programmeur devrait avoir de bonnes capacités analytiques, logiques et être scrupuleusement précis. D'autre part, ils retiennent de C. C. Upshall et L. H. Riland (1958), de la société Kodak, la nécessité d'avoir la capacité à travailler sous pression, en équipe, à avoir une mémoire rémanente, à s'adapter aux changements et à voir un problème lié aux compétences mathématiques. À partir de ces constats, ils élaborent un premier test, d'abord intitulé EDPM (*Electronic Data Processing Machine*), puis PAT (pour *Programming Aptitude test*). Il s'agit d'un test d'une heure, qui contient plusieurs parties, dont des séries de nombres, des analogies de figures et des raisonnements arithmétiques. En cela et de l'aveu même de Walter McNamara, il est en fait assez

5 On peut citer le Logical Analysis Device, le Thurstone Primary Mental Abilities Test, le Gordon Personal Profile, le Gordon Personal Inventory, le Thurstone Temperament Schedule et le Edwards Personal Preference Schedule.

proche de l'*ACE Psychological Examination*⁶, à la différence près que les délais pour chaque section ont été rallongés afin d'accroître la fiabilité et qu'on a rendu un certain nombre d'éléments plus difficiles.

L'efficacité de ces tests est également contestée. En effet, au sein de la principale société savante informatique, l'ACM (*Association Computer Machinery*) est créé un sous-groupe qui s'intéresse principalement à la gestion du personnel, le « *Special Interest Group on Computer Personnel Research* » (SIGCPR). Ce groupe mène des réflexions et produit des expertises dont de nombreuses se concentrent sur la mesure de l'efficacité. On rend notamment compte de travaux qui interrogent la question de la capacité à prédire des recrutements. Si certaines études tentent de montrer l'efficacité de ces tests, elles ne font pas l'unanimité au sein de ce groupe. On y montre notamment que les résultats sont finalement très liés à la qualification (le niveau de diplôme) (Alspaugh, 1971), au secteur de qualification (Reinstedt, 1967) ou encore à l'âge des individus. Ainsi, pour Bairdain (1964), les moins de 25 ans ont de bien meilleurs scores et la performance aux tests décroît rapidement avec l'âge⁷. À

6 « *This is a one-hour test consisting of sections on number series, figure analogies, and arithmetic reasoning. It contains the same type of items found in the & part of the ACE Psychological Examination* », Upshall C. & Riland N. (1958). « An unpublished study at Eastman Kodak Company » (cité in McNamara & Hugues, 1961, p. 41).

7 Même si là encore, ce n'est pas l'âge en soit qui importe mais davantage la durée écoulée depuis le dernier diplôme.

l'inverse, il est difficile de faire des liens entre la réussite à ces tests et d'autres types d'aptitudes. C'est notamment le cas pour les aptitudes à la créativité (Perry & Cannon, 1967). Wolf (1971) tente quant à lui de lier la réussite de ces tests à la profession des individus, sans pour autant obtenir des résultats significatifs et en vient ici à défendre que dans chacune des professions, on peut trouver des personnes capables de les réussir.

Les fondateurs de ces tests, notamment Walter McNamara, précisent qu'il ne s'agit pas tant de comprendre quels sont les déterminants de la réussite de ces tests, que de prédire la réussite professionnelle dans le secteur grâce aux tests du PAT, dont les résultats sont donc étroitement corrélés à la performance future en situation professionnelle. Ce point est, de fait, plus difficile à mettre en évidence d'un point de vue expérimental et ne peut se faire qu'*a posteriori*. Concernant le PAT, un certain nombre de commentateurs avançaient que même si cette évaluation pouvait avoir lieu, on pouvait légitimement mettre en avant le côté auto-renforçant de ces évaluations. En effet, ceux qui pouvaient mesurer ces résultats de performance dans les organisations ont fort probablement eux-mêmes été sélectionnés selon le principe du test PAT. Ainsi, jusqu'au début des années 1980, le SIGCPR a poursuivi l'étude de ces tests, sans que leur efficacité ne soit réellement démontrée⁸.

Un autre point objectif de remise en cause de ces tests qui se veulent impartiaux et objectifs concerne la question de l'interculturalité. En effet, au sein des firmes et organisations, les mêmes tests sont diffusés dans le monde entier, sans aucune adaptation selon le pays ou la zone géographique concernée. On apprend par exemple que le test IBM-Pat a été traduit en allemand, sans que ne soit réalisée une seconde évaluation des tests. On retrouve la même situation dans leur adaptation en Afrique du Sud (Hall, 1970). D'autre part, et toujours sur ce point, assez rapidement des aspects légaux viennent se mêler de l'utilisation de ces tests. En effet, aux États-Unis, une loi d'août 1966, issue des travaux de l'*Equal Employment Opportunity Commission* demande à ce que tous les éditeurs de tests assurent qu'ils ne comportent pas de biais « *raciaux ou culturels* ». En effet, d'autres tests d'intelligence générale largement utilisés, comme le *Wonderlic Test*, ont été largement critiqués dans une étude, car contenant des biais racistes (Ruda & Albright, 1968). Or, les tests de programmation, aussi impartiaux pouvaient-ils se présenter, n'étaient pas à l'abri de ce type de critiques.

Au-delà de cette critique morale et légale, les limites interculturelles engendrent également des blocages organisationnels, préjudiciables pour l'en-

⁸ « *It is also not possible to reliably predict success in learning programming on the basis of normally*

observable external personal attributes or standardized written tests when the people involved are of college level » (Mazlack, 1980).

treprise. Ainsi, en 1970, Geert Hofstede, par la suite devenu l'une des figures majeures du management interculturel, était à l'époque employé d'IBM, en charge du recrutement pour la zone Europe, Afrique et Moyen Orient. Il a notamment été chargé de prendre en compte un problème apparu assez rapidement dans l'utilisation des tests PAT pour le recrutement des programmeurs en Afrique. On y observait que le pourcentage de réussite à ces tests était très faible⁹. Il s'avéra que plus que la question du test lui-même, c'était davantage l'usage de la langue et l'habitude de passer des tests qui eut un impact sur ces scores. Geert Hofstede, proposa un test adapté, prenant en compte ces différents aspects « interculturels ». Ce nouveau test fut très mal perçu par les dirigeants africains d'IBM qui souhaitaient que soit conservé le test originel américain. Au-delà de la remarque de Hofstede sur la fierté locale il faut également y voir un décalage entre le test en soi et son usage. Cette anecdote illustre assez bien l'ambivalence présente autour de l'usage ces tests. Face à l'impératif du recrutement, une sélection est nécessaire : ce n'est donc pas tant le contenu du test qui importe que sa fonction de sélection. Ensmenger (2010), qui, interrogeant le cas de SDC, pose explicitement la question : si ces tests ne fonctionnent pas, pourquoi poursuivre leur utilisation ? Pour SDC, la réponse

est simple : l'entreprise était engagée par un contrat dans le développement de logiciel et devait recruter un grand nombre de programmeurs¹⁰. À travers notre revue de la littérature, ce constat peut, en quelque sorte, être généralisée à l'ensemble des firmes ayant utilisé ces tests à cette époque.

Du côté français, déjà au début du xx^e siècle, les études de Munsterberg (1913) ou de Lahy (1913) développaient des programmes de recherche sur l'utilisation de tests chez les conducteurs de tramway ou les dactylographes. Pour ce qui deviendra l'informatique, les tests avaient été assez tôt introduits pour les professions qui précèdent l'émergence de l'informatique (Neumann, 2013). C'est ainsi qu'en France les tests de sélection dans l'industrie ont été principalement utilisés pour la sélection des dactylographes et des mécanographes, professions dont les tâches sont les plus répétitives. Ces tests adaptaient à la marge des tests psychotechniques déjà utilisés par d'autres professions techniques (*ibid.*, p. 94), dans la filiation psychométrique française.

Pour l'informatique à proprement parler, au sein de la littérature scientifique, on ne retrouve pas de trace de débat sur l'utilisation des tests. Il n'existe pas de société savante sur la question et comme l'indique Mounier

⁹ Cette expérience est largement relatée pages 271-272 dans l'ouvrage de management de Hofstede & Minkov (2010).

¹⁰ Ensmenger montre par ailleurs que ce choix a eu un fort impact sur la structure des projets logiciel (2010, p. 63).

Khun, on s'intéresse davantage aux questions purement scientifiques concernant la discipline et beaucoup moins à la question de la gestion du personnel (Mounier-Khun, 2010)¹¹. Cette question est néanmoins davantage présente chez les praticiens et est au cœur de leurs préoccupations. Ces derniers semblent être à l'initiative de l'importation des tests américains, qui selon eux s'adaptent bien à la profession de programmeur. Comme le précise Neumann, à cette époque pour les praticiens « L'élargissement des notions de système et d'information conduit à minorer l'importance des aspects techniques et à mettre en avant celle des problèmes d'organisation » (Neumann, 2013, p 208). C'est notamment le cas de Gérard Bauvin, l'un des premiers en France à théoriser la question de la gestion des équipes dans l'informatique. Il est alors directeur de Cegos Informatique. Il s'agit pour lui de repérer des aptitudes chez les programmeurs, telles que l'esprit d'équipe ou le soin du détail. Selon lui, « *la formation de base n'a pas de grande influence sur la qualité du programmeur (limité à des fonctions de programmation), ce sont avant tout des praticiens* » (Bauvin, 1968, p. 316). Pour leur sélection, il préconise ainsi de faire passer les tests proposés par les constructeurs. Pour

la plupart, il s'agit du test PAT, même lorsqu'ils n'utilisent pas les machines IBM. Bauvin argumente que cette sélection fonctionne à condition d'être stricte. Chez CEGOS, « *moins de 5 % des candidats sont retenus sur la base de ces tests* » (*ibid.*, p. 317). Dès les années 1970, la question est présente, et sera reposée régulièrement jusqu'aux années 1990. Sur cette période, on observe néanmoins que chercheurs et praticiens sont beaucoup moins enthousiastes sur cette question que leurs prédécesseurs.

Utilisation différenciée des tests et hiérarchisation des formations et des publics

Que ce soit aux États-Unis ou en France, on observe au fil du temps de fortes différences dans l'utilisation de ces tests, ce qui se traduit par l'absence d'utilisation de tests spécifiques dans l'enseignement supérieur, et à l'inverse un large usage par les formations privées, qui en font une utilisation opportuniste et commerciale. Ceci vient acter une hiérarchisation des formations, entre le supérieur et le privé. Plus tardivement, l'utilisation de ces tests disparaîtra de la sélection par les entreprises. On pourrait y lire l'effet de la réhabilitation de la profession dans le champ, mais également le résultat d'un débat propre au à la psychométrie qui aboutit à la victoire du « paradigme clinique ».

¹¹ En effet, au sein des sociétés savantes de l'époque, qu'il s'agisse de l'Association Française de Calcul (AFCAL) créée en 1957 ou de la société française de recherche opérationnelle (SOFRO) créée en 1956, les questions de sélection du personnel personnels ne sont pas abordées.

Une non-utilisation dans l'enseignement supérieur aux États-Unis comme en France

Aux États-Unis, accompagnant le développement d'IBM, ces tests sont très largement utilisés, puis diffusés et adoptés par les entreprises du secteur émergent de l'informatique. L'usage des tests dans le secteur est assez massif, puisqu'en 1966 aux États-Unis, 25 000 personnes ont été recrutées via ces derniers. En 1966, Dickmann et Lockwood, à partir d'un sondage auprès de 483 firmes américaines montrent que 72 % des firmes du secteur informatique utilisent le PAT. Ce taux est de 68 % pour le Canada. Les autres tests tels que le *Wonderlic* ou le *Thurstone Test of Mental Alertness* ou encore *Otis* qui sont des tests d'intelligence générale –, le *Thurstone Temperament Schedule*, *Activity Vector Analysis* – qui sont davantage des tests de personnalité – sont plus marginalement utilisés.

Pour l'enseignement, et plus particulièrement l'enseignement supérieur, il en est autrement. Alors que l'enseignement de l'informatique tend à s'institutionnaliser, ces tests y occupent une place marginale (Gupta, 2007). En effet, si le milieu universitaire américain de l'enseignement informatique adopte bien des tests, ces derniers ne sont pas spécifiques à l'informatique et on retrouve pour l'entrée dans les formations l'utilisation du SAT, passage obligé et « *totem* » (Lemman, p. 273) pour les familles américaines qui sou-

haitent que leurs enfants suivent des études supérieures.

Avant tout marqué par la loi Faure d'après 1968 portant sur la non-sélection pour l'université et sur le maintien des classes préparatoires aux grandes écoles (Jalabert, 2016), le milieu universitaire français n'adopte pas du tout ces tests, qu'ils soient généraux ou spécifiques. Excepté pour le TOEFL, le test d'anglais développé par ETS et largement utilisé dans les écoles d'ingénieurs ou de commerce, l'enseignement supérieur français est donc « à l'abri » de la généralisation, voire de l'usage des tests de sélection. Ce n'est pas le cas des écoles de commerce qui les utilisent assez tôt, à l'instar de l'Institut supérieur des affaires d'HEC qui est un troisième cycle de l'école de commerce et qui recrute des diplômés du supérieur après une « *épreuve de synthèse d'un dossier, et des tests psychologiques* »¹².

Pour l'informatique en France, l'une des premières formations universitaires à apparaître dans le paysage est celle de l'Institut de programmation (Mounier-Khun, 2010), avec la formation intitulée « *Opérateurs et programmeurs de calculatrices électroniques* ». Il s'agit de « *donner leur chance à des jeunes gens qui n'ont pu s'adapter à l'enseignement secondaire, mais se révèlent aptes à la programmation* » (Mounier-Khun, 2010, p. 284).

12 « L'Institut supérieur des affaires ouvrira à la rentrée 1969 », 21 avril 1969, *Le Monde*.

Il n'y a donc pas de diplôme requis, ni de passage de test *a priori*. « *Aucun diplôme n'est exigé. L'enseignement est limité à quelques matières, pendant le premier mois. Les étudiants passent, à l'issue de cette période préparatoire, un examen probatoire qui effectue une certaine sélection. La durée des études est de six mois* »¹³.

Par la suite, les autres formations sélectionnent sur la base de diplômes, principalement le baccalauréat pour les formations universitaires. À la création des Maîtrises d'Informatique Appliquées à la Gestion des Entreprises (MIAGE) il est dans un premier temps envisagé de réaliser une sélection sur tests. « *Les étudiants devront avoir obtenu un diplôme de premier cycle, mais pas nécessairement scientifique ; c'est le conseil de l'établissement qui fixera les modalités d'admission (sur dossiers ou sur tests, par exemple)* »¹⁴. Finalement, la création des maîtrises MIAGE en 1970 impliquera bien une sélection, mais cette dernière se fera sur un modèle calqué sur ce qui est appliqué dans les Instituts Universitaires de Technologie (IUT), à savoir une sélection à l'entrée basée sur le diplôme et l'examen du dossier, (Mercuroff, 2015). C'est-à-dire une sélection académique classique où la question des aptitudes propres à l'informatique est exclue.

¹³ « La formation des opérateurs et programmeurs de calculatrices électroniques », 25 janvier 1965, *Le Monde*.

¹⁴ « Création d'une maîtrise d'informatique de gestion », 30 septembre 1970, *Le Monde*.

2.2 Un usage opportuniste et commercial des tests dans les formations privées dans les années 1970 en France

Ce sont en revanche les formations privées qui vont faire un usage systématique de ces tests. Dès l'apparition de l'informatique dans les organisations, et avant même qu'un enseignement universitaire et scolaire ne soit structuré, de nombreuses formations privées ont émergé dans les années 1960, et se sont largement diffusées au cours des années 1970. Ces formations vont alors utiliser des tests de recrutement comme sorte d'examen d'entrée. Pour la plupart ce ne sont pas des tests de programmation du type PAT mais de simples tests que les organismes de formation conçoivent eux-mêmes.

Il est intéressant de comprendre quel usage il en est fait : en effet, ils acquièrent assez rapidement une vocation commerciale, et visent à attirer des potentiels candidats aux formations. On peut ainsi trouver de nombreuses annonces de ce type :

Programmeur : un des métiers les mieux payés, des débouchés partout (lisez les offres d'emploi). Ce nouveau métier s'apprend en cours du jour ou du soir avec des professeurs et un ordinateur. Aucun diplôme exigé (le niveau BEPC est souhaitable). À votre attention nous avons établi un test gratuit pour qu'il vous renseigne sur vos possibilités¹⁵.

¹⁵ Encart publicitaire 12 janvier 1983, *01.informatique*.

Le style de ces annonces s'oppose d'une part aux savoirs scolaires et leur validation par les diplômés (ces tests s'adressant à des personnes qui ne sont pas ou peu diplômées) et d'autre part à la non-reconnaissance des aptitudes et des compétences des individus au sein des entreprises. Par exemple : « *vous n'êtes pas aussi bête que votre patron le pense* » (Encart publicitaire pour Control Data novembre 1971, le nouvel observateur) et de compléter « *le patron barre la route de toute promotion, sous prétexte que vous n'en avez pas la capacité* » (*ibid.*).

Ce recours aux tests n'a pas une fonction de barrière à l'entrée, comme on a pu l'observer pour le recrutement, mais au contraire une fonction commerciale qui vise à élargir les publics et devient même un élément déterminant de la stratégie commerciale de ces formations. Un article de 1982 critique la méthode utilisée :

Accompagné d'un « conseiller d'orientation » le futur élève pénètre dans la salle de cours où des stagiaires s'initient à la programmation sur écran. C'est le choc : pourquoi pas moi ? Retour dans un bureau : proposition de test dont la durée varie entre 20 et 30 minutes. Quelques instants d'attente angoissée... Documents en main, le conseiller – tout sourire – annonce aux postulants que ses résultats sont favorables, qu'il reste encore deux places disponibles au prochain cours. Comment résister à une telle offre de formation débouchant dit-on sur un emploi¹⁶ ?

Cette méthode a donc pour finalité d'obtenir la signature d'un contrat et fait partie d'un démarchage plus global qui s'adresse à un large public. Ces formations s'adressent en effet à un public peu ou pas diplômé et sont perçues donc comme une « seconde chance » pour les candidats, pour qui la scientificité apparente des tests rassure et pour qui ces tests sont censés garantir l'objectivité de leur recrutement. « *Pour cette population traumatisée par l'échec, [...] les tests rassurent, les gens sont tellement heureux de réussir à des tests [...] ils n'en avaient plus l'habitude* » (*ibid.*)

Les tests avaient donc ici précisément cette fonction de revalorisation des individus et de leur capacité à s'engager dans une démarche de promotion sociale. Malheureusement, ce secteur a mauvaise réputation et est rapidement présenté comme un « *petit monde secoué par des scandales retentissants* » (*01.informatique*, 18 mai 1990). En effet, un certain nombre de ces entreprises de formation se sont révélées être de mauvaise qualité. Déjà au début des années 1980, on s'inquiétait du développement de ces écoles : « *beaucoup de cours, peu de véritables écoles* »¹⁷. Elles ont fait l'objet, entre 1970 et 1990, d'un long contentieux au niveau juridique : « *Faut-il casser les écoles privées du secteur informatique ?* » (*01.informatique*, 13 février 1983). En effet, même si une loi du 12 juillet 1971 interdit le démarchage pour ce type de formation, il aura fallu de nombreuses années d'ac-

16 *01.informatique*, 12 avril 1982

17 *01.informatique*, 19 avril 1982.

tions au niveau institutionnel pour que ces pratiques cessent (Vicente, 2018). C'est donc dans ce contexte que les tests se sont retrouvés être des outils de cette duperie.

L'abandon des tests dans les organisations : réhabilitation de l'informaticien et victoire du « paradigme clinique »

Ce décalage dans l'utilisation des tests par l'industrie et les formations privées d'un côté et l'enseignement supérieur de l'autre pourrait être perçu comme une accentuation de la différence entre les métiers répétitifs et sous-qualifiés d'un côté et les qualifiés de l'autre. Les premiers devant être testés et mesurés par la psychométrie, pour les autres se destinant à des emplois plus qualifiés ont déjà été sélectionnés par le système éducatif. On retrouve dans cette dichotomie celle opérée entre ce qui est mesurable, évaluable par les tests et ce qui est de l'ordre de l'incommensurable (Marchal, 2015, p. 267) et qui doit faire appel à d'autres méthodes.

À la fin des années 1970, on recommande de manière explicite de ne pas les faire passer aux personnes diplômées de niveau universitaire « *car les résultats sont dénués de sens et pourraient nuire aux candidats* » (01.informatique, 29 février 1979).

Plus globalement la presse se fait l'écho de l'inefficacité des tests. En 1979 par exemple, est fait référence au titre

d'une publication, « Prédire les résultats des étudiants à un cours d'initiation à la programmation »¹⁸, parue dans le *Computer Journal* en novembre 1978 (Maziack, 1978), l'organe de la British Computer Society. Lewis Maziack, un chercheur canadien du département de calcul et informatique de l'Université de Guelph (Ontario), entend aller contre le sens commun selon lequel il y aurait des aptitudes, un don particulier ou une forme d'esprit particulière propre aux programmeurs que les tests seraient censés détecter. En réalisant une étude autour de la passation du test PAT d'IBM, il montre qu'on ne trouve pas de corrélation entre le fait de suivre des cours de programmation et la réussite à ces tests : « *en somme pour choisir un programmeur, les tests proposés ne paraissent guère plus fiables que la méthode qui consiste à mouiller son doigt pour savoir d'où vient le vent* » (01.informatique 29 février 1979).

Vers une victoire du paradigme clinique ?

Ces critiques semblent par la suite avoir été prises en compte dans les pratiques de recruteurs. On observe que ces dernières rentrent en concurrence avec d'autres méthodes qui deviennent hégémoniques, et en même temps qu'elles accompagnent le recrutement. Ces nouvelles méthodes illustrent ce que Thomas Le Bianic nomme « *la place*

¹⁸ 01.informatique, 26 février 1979.

croissante occupée par le paradigme clinique » (2004, p. 209). Selon lui, ce tournant dans l'histoire de la psychométrie se situe à la fin des années 1970, où dans des institutions comme le Cnam on observe le développement de méthodes qui portent leur analyse sur d'autres éléments que le travail à proprement parler et un décalage croissant entre les modèles théoriques et leurs applications pratiques : « *L'analyse fine du travail, sur laquelle se fondait la pratique des psychophysiologistes du travail de l'entre-deux-guerres, est délaissée par les praticiens au profit de tests ou de techniques psychosociologiques standardisées, véhiculées d'une entreprise à l'autre* » (*ibid.*, p 203). La « graphologie » et des tests de personnalité sont de bonnes illustrations de cette victoire. Selon Emmanuelle Marchal (il s'agit par ces tests de déceler les « *qualités morales des candidats* » (2005, p. 64), plus que leurs compétences techniques).

Et c'est effectivement ce qu'on observe dans l'industrie informatique. Ainsi, en février 1982, les pages Emploi proposent de décrire les rouages du recrutement dans le secteur. À partir d'une enquête faite auprès des recruteurs, *01.informatique* propose un panorama des pratiques de recrutement des entreprises du secteur, qu'il s'agisse des constructeurs, des sociétés de service ou des cabinets de recrutement. On y décrit alors le poids accordé aux différentes composantes qui pèse dans le recrutement : le *curriculum vitae*, la lettre de motivation, l'entretien et les tests.

On y apprend ainsi que les lettres de motivation ont une faible importance, que le CV est avant tout là pour viser l'adéquation au profil et que ce sont les entretiens, jusqu'à cinq pour des entreprises comme IBM, qui sont les plus déterminants dans le processus de recrutement. Paradoxalement, les tests du type de ceux que nous avons décrits sont extrêmement peu présents.

Ainsi, même chez IBM, on va préférer des petits tests logiques, un test d'anglais, voire la graphologie. Chez Compagnie internationale pour l'informatique Honeywell-Bull (CII-HB)¹⁹ on utilise une « *graphologie rapide* » ainsi qu'un test « maison » de personnalité. De la même manière, les cabinets de recrutements n'utilisent pas ces tests de programmation, ils disent utiliser de manière « *facultative* » ou « *exceptionnelle* » des tests de graphologie, psychologiques ou de comportement. Ce sont les sociétés de service, et notamment la Compagnie Générale d'Informatique (CGI) qui utiliseront davantage une batterie de tests présentés comme « *déterminants* » ; il s'agit principalement de « *tests psycho-techniques de raisonnement* » ou de tests de logique qu'ils disent également mobiliser en complément d'une « *analyse graphologique systématique* ».

On voit donc que ce sont les sociétés de services qui s'attachent le plus aux

¹⁹ Initialement créé en 1966 dans le cadre du Plan Calcul, la Compagnie internationale pour l'informatique a été absorbée par la société Honeywell-Bull en 1975.

tests, et que finalement les cabinets de recrutement qui sont censés s'appuyer sur ces procédures utilisent un plan beaucoup vaste qui correspond davantage aux pratiques de ressources humaines que l'on retrouve à l'époque.

La mise à l'écart de l'usage systématique des tests informatiques pour d'autres tests et d'autres méthodes participe également à la valorisation du métier de recruteur, qui met de son côté en place une véritable expertise. « *Je suis dans le métier depuis 1959, j'ai travaillé avec des tests, on les utilise d'autant plus que l'on a moins d'expérience, un entretien bien conduit donne des résultats tout aussi satisfaisants qu'un test qui la plupart du temps ne fait que confirmer notre diagnostic* »²⁰.

Ces tests seraient donc essentiellement adressés aux débutants qui n'ont pas une vraie vision du métier de recruteur. De leur côté, les cabinets de recrutement se disent partisans d'une réintroduction du côté humain dans le processus. En effet, ces derniers tests créent une véritable controverse dans la presse professionnelle, ainsi que l'affirme l'un des recruteurs : « *les tests à caractère projectifs devraient être réservés à une application clinique, leur utilisation comme instrument de recrutement est abusive, nos interlocuteurs ne sont pas des cobayes, ce sont des hommes* » (*ibid.*). Ces tests incarnent ici la face déshuma-

nisante du recrutement, en se concentrant sur la mesure des aptitudes²¹.

Signe de la valorisation du métier, quelques années plus tard, en 1989, on parle alors de « *chasseurs de têtes* » pour évoquer ceux qui sont en charge de recruter ces informaticiens. Parmi les outils utilisés, ce sont la graphologie et les entretiens qui tiennent le « hit-parade » (*01.informatique*, 20 février 1989) des outils de recrutement. On reproche notamment aux tests la difficulté de leur mise en place. « *Les tests interviennent peu chez Ordis, tout simplement parce que c'est une procédure lourde* » (*ibid.*) et pour d'autres c'est encore leur efficacité qui est questionnée. Chez Sema-sélection (Société d'économie et de mathématiques appliquées) qui recrute essentiellement dans les grandes écoles, « *nous n'employons pas de tests projectifs, et ce à dessein. Aucun test n'est fiable à 100 % et leurs résultats ne peuvent que donner des éclairages et permettre d'aller plus en avant sur des points particuliers à la condition d'en discuter en tête à tête avec l'intéressé. En général, ils offrent beaucoup plus l'occasion d'approfondir un entretien* » (*ibid.*).

À la fin des années 1990, les tests propres à l'informatique semblent avoir disparu. Dans une étude publiée en 2000, Christian Bessy, Guillemette de Larquier et Marie-Madeleine Vennat étudient le marché du travail des informaticiens,

²⁰ 22 février 1982, *01.informatique*.

²¹ Sur la critique de la mesure de l'homme, voir Jay Gould (1981).

à partir de l'étude des offres d'emploi. Dans leur comparaison entre la France et la Grande-Bretagne, les tests propres à l'informatique ne sont plus du tout présents, et selon les auteurs « *les annonces françaises accordent plus d'importance à l'envoi d'une lettre de candidature (1^{er} rang), ce qui peut être relié aux méthodes de sélection des candidats, notamment au recours en France à la graphologie et, plus généralement, aux méthodes d'évaluation de la personnalité autres que les tests* » (Bessy, de Larquier & Vennat, 2000, p. 17), marquant ainsi définitivement la préférence pour les diplômes, mais également la victoire du paradigme clinique pour les recrutements.

risation de l'informaticien, cadre d'entreprise. Plus qu'une différence entre la France et les États-Unis, ou encore entre le monde de la formation et celui des entreprises, ce qui se dégage est précisément une forte différence entre le public et le privé. L'histoire de l'informatique se situant précisément à l'interface de ces deux modalités, il est donc normal que la formation à l'informatique se retrouve traversée par ces tensions.

Conclusion

Créés dans un premier temps afin de répondre à une fonction sociale de sélection à l'entrée d'une nouvelle industrie pour laquelle il n'y a pas ou encore peu de formations universitaires reconnues, les tests propres à l'informatique ont connu une trajectoire singulière. Nous avons ainsi pu voir qu'au cours du temps, il en est fait un usage différencié en fonction des acteurs impliqués. Ainsi, si l'utilisation de ces tests résiste dans le temps, c'est qu'ils ne remplissent que partiellement leur fonction sociale de sélection, de barrière à l'entrée, et d'attractivité pour la profession. Au fur et à mesure que se structure la formation, notamment universitaire, les tests disparaissent. On voit en filigrane dans le rejet de ces tests la valo-

Bibliographie

Alsbaugh J. (1971). « The Relationship of Grade Placement to Programming Aptitude and FORTRAN Programming Achievement ». *Journal for Research in Mathematics Education*, vol. 2, n° 1, pp. 44-48.

Bessy C., De Larquier G. & Vennat M.M. (2000). « Le marché du travail des informaticiens médiatisé par les annonces d'offres d'emploi : comparaison France/ Grande-Bretagne ». Dossiers de recherche 05, Centre d'études de l'emploi.

Bairdain J. (1964). « Research studies of programmers and programming ». Unpublished studies, pp. 78-136.

Canning R. (1957). *Installing Electronic Data Processing Systems*. New York : John Wiley & Sons.

Dickmann R.A. & Lockwood J. (1966). « Survey of Test Use in Computer Personnel Selection. Technical Report ». Proceedings of the Fourth SIGCPR Conference on Computer Personnel Research. New York : ACM Press, pp. 61-82.

Garric D. (1972). *L'homme électribal ou la civilisation de l'ordinateur*. Paris : Grasset.

Gould J. (1981). *La Mal-mesure de l'homme : l'intelligence sous la toise des savants*. Paris : Odile Jacob.

Gupta G. (2007). « Computer science curriculum developments in the 1960s ». *IEEE Annals of the History of Computing*, vol. 29, n° 2, pp. 40-54.

Hall R.S. (1970). « The Construction Of A Selection Battery For Programmers Adapted To South African Conditions ». SIGCPR'70 Proceedings of the eighth annual SIGCPR conference. College Park (MA) : ACM, pp. 108-118.

Hofstede G. & Minkov M. (2010). *Cultures*

and Organizations Software Of The Mind Intercultural Cooperation and Its Importance for Survival. New York/London : McGraw-Hill.

Huteau M. (1996). « L'école française de psychotechnique 1900-1940 ». In Y. Clot (éd.). *Les Histoires de la psychologie du travail*. Toulouse : Octarès.

Jalabert L. (2016). « Transformer l'université. La politique pour l'enseignement supérieur avant la loi Faure (1958-1968) ». In B. Poucet & D. Valence (dir.). *La loi Edgar Faure. Réformer l'université après 1968*. Rennes : PUR.

Lahy J. (1913). *Étude des signes physiques de la supériorité professionnelle chez les dactylographes*. Paris : Ed. Masson.

Lahy J. (1913) *La Supériorité professionnelle chez les conducteurs de tramways dans ses rapports avec la consommation d'énergie électrique*. Paris : H. Dunod et E. Pinat.

Le Bianic T. (2004). « Le Conservatoire des Arts et Métiers et la « machine humaine ». Naissance et développement des sciences de l'homme au travail au CNAM (1910-1990) ». *Revue d'Histoire des Sciences Humaines*, vol. 11, pp. 185-214.

Le Bianic T. (2008). « Des tests pour les chômeurs : la psychotechnique au ministère du Travail, des années 1930 aux années 1950 ». *Bulletin de psychologie*, vol. 4, n° 496, pp. 357-366.

Lemann N. (1999). *The Big Test : The Secret History of the American Meritocracy*. New York : Farrar, Straus & Girou.

Marchal E. (2005). « Graphologie et entreprises : histoire et controverses ». *Sociologies pratiques*, vol. 1, n° 10, pp. 57-76.

Marchal E. (2015). *Les embarras des recruteurs. Enquête sur le marché du travail*. Paris : Éd. de l'EHESS, coll. « Cas de figure ».

Maziack L. (1978). « Predicting Student

Success in an Introductory Programming Course ». *Computer Journal*, vol. 21, n° 4, pp. 380-384.

Maziack L. (1980). « Identifying Potential to Acquire Programming Skill ». *Communications of the ACM*, vol. 23, n° 1, pp 14-17.

McNamara W. & Hughes, J. (1961). « A Review of Research on the Selection of Computer Programmers ». *Personnel Psychology*, 14, n° 1, pp. 39-51.

Mercoureff W. (2015). « L'informatique dans l'enseignement et au CNRS au début des années soixante-dix ». *Histoire de la recherche contemporaine*, vol. 4, n° 2, pp. 180-185.

Münsterberg H. (1913). *Psychology and industrial efficiency*. Boston : Mifflin.

Neumann C. (2013). « De la mécanique à l'informatique : les relations entre catégorisation des techniques, groupes professionnels et transformation des savoirs managériaux ». Thèse de doctorat en histoire, Paris-Ouest Nanterre la Défense.

Perry D. & Cannon W. (1967). « Vocational Interests of Computer Programmers ». *Journal of Applied Psychology*, vol. 51, n° 1, pp. 28-34.

Pinell P. (1995). « L'invention de l'échelle métrique de l'intelligence ». *Actes de la recherche en sciences sociales*, vol. 108, pp. 19-35.

Reinstedt N. (1967). « Results of a Programmer Performance Prediction Study ». *Transactions on engineering management*, vol. 14, n° 4, pp. 183-187.

Rowan T. (1958). « The Recruiting and Training of Programmers ». *Datamation*, vol. 4, n° 3, pp. 16-18.

Rowan TC (1957). « Psychological Tests and Selection of Computer Programmers ». *Journal of the ACM*, vol. 4, n° 3, pp. 341-347.

Ruda L. & Albright E. (1968). « Racial differences on selection instruments related to subsequent job performance ». *Personnel Psychology*, vol. 21, n° 1. pp. 31-42.

Trépo G. & Estellat N. (2002). *L'appréciation du personnel mirage ou oasis ? Paris : Éditions de l'organisation*.

Vettrains-Soulard M.-C. (1983). « La presse informatique et ses publicités ». *Communication et langages*, n° 56, pp. 99-108.

Vicente M. (2018). « Les dérives commerciales des organismes privés de formation à l'informatique et leur contrôle (1971-1990) ». *Éducation permanente*, n° 215, pp. 209-220.

Wolf L. (1971). « Perspectives on Testing for Programming Aptitude ». *ACM'71 Proceedings of the 1971 26th annual conference*. New-York (NY) : ACM, pp. 268-277.

« Si quelqu'un m'explique, je vais pas comprendre. » Une formation d'informatique face à la pédagogie de l'autonomie

Mathilde Krill

IRISSO, Université Paris-Dauphine / PSL

Résumé

À partir d'un travail ethnographique de plusieurs mois, cet article analyse la mise en pratique d'une pédagogie « innovante » – qui met l'autonomie au centre de l'apprentissage – dans une formation qui vise à transformer des demandeur-ses d'emploi en développeur-ses. Dans un premier temps, l'article décrit la réception de la pédagogie active par les participant-es de la formation – tant les « apprenant-es » que les encadrant-es. Dans un second temps, l'article met au jour les différents prérequis invisibles tout au long de l'apprentissage du code dans cette formation.

Mots-clés : sociologie de l'éducation ; formation en informatique ; pratique pédagogique ; demandeurs d'emploi ; autonomie.

Abstract

Drawing on ethnographic fieldwork conducted over several months during a training program that aims to turn job seekers into developers, the article examines how participants implement an 'innovative' pedagogy which puts autonomy at the core of the learning process. First, the article describes the ways learners and supervisors react to active pedagogy. Secondly, the article reveals the various invisible prerequisites that structure the ways participants learn code during the program.

Keywords: sociology of education; coding bootcamp; teaching practice; job seekers; autonomy.

Introduction

« Apprendre à apprendre » : c'est l'une des idées au cœur des « pédagogies innovantes » valorisées dans certaines écoles d'informatique¹. Dans ce domaine, ce mantra se verrait notamment justifié parce qu'il faut que tout-e professionnel-le puisse s'adapter aux évolutions technologiques et soit donc en mesure d'apprendre par lui-même toute sa vie. Ainsi, dans la formation en développement informatique pour personnes en recherche d'emploi qui sera l'objet de cet article, l'« autonomie » de chacune est à la fois un prérequis et un objectif, une attente formulée régulièrement et une nécessité quotidienne. Il n'y a plus d'enseignant-e, il n'y a plus d'élève, il y a des « apprenant-es », réuni-es ensemble dans une salle, qui, lorsqu'ils souhaitent apprendre quelque chose, doivent le chercher sur Internet.

Ces injonctions à l'autonomie ne touchent pas que les formations en informatique. De nombreux travaux se sont intéressés à l'autonomie en école primaire, soulignant qu'elle était depuis des décennies devenue un objectif de l'école (Lahire, 2001 ; Raab, 2016). Certains se sont intéressés à la mise en pratique de l'autonomie – qui se réduit souvent à du travail individuel (c'est-à-dire des exercices faits seul) (Lahire, 2001) – ou, plus

généralement, à la mise en pratique de valeurs néolibérales ayant pénétré l'école, en incluant l'autonomie (Clément, Dreux, Laval & Vergne, 2011 ; Dürler, 2015). De plus, les récentes évolutions du système scolaire tendent à laisser une place prééminente à ce que Bernstein appelle le code intégré (Bonnéry, 2009a ; Duru-Bellat, Farges & Zanten, 2018). Le code intégré est une manière d'organiser un programme d'enseignement opposée au code sériel : dans le code intégré, l'enseignement est organisé autour d'un projet qui mobilise différentes disciplines et met en avant la manière d'apprendre plus que le contenu des savoirs enseignés, au contraire du code sériel, dans lequel les savoirs sont bien distingués et une discipline est imposée aux élèves (Bernstein, 1975). Le code sériel est associé à un enseignement traditionnel alors que les pédagogies nouvelles relèvent du code intégré. Certaines recherches récentes établissent que le code intégré se retrouve notamment dans les systèmes éducatifs de l'Europe nordique et de pays anglo-saxons (Mons, Duru-Bellat & Savina, 2012). Le code intégré est, selon les travaux de Bernstein, plutôt associé à une « *pédagogie invisible* », où les attendus d'apprentissage ont tendance à être implicites (Bernstein, 1975).

Du côté de la formation continue, le constat est aussi fait d'une individualisation et d'une responsabilisation des apprenant-es dans leurs parcours par le biais de formations numérisées (Benedetto-Meyer, Boboc & Metzger 2019). En particulier, Boboc et Metzger

¹ Voir par exemple sur le site de l'école d'informatique fondée par Xavier Niel, 42 : « La méthode 42 repose sur "apprendre à apprendre" » [URL : <https://42.fr/42-cest-quoi/la-methode-42/>] (s.d.).

ont travaillé sur une formation numérique suivie à distance par des salarié-es en marge de leur temps de travail : les salarié-es doivent alors s'organiser individuellement pour trouver du temps chaque semaine à consacrer au suivi d'un cours en ligne conçu par leur entreprise. Certain-es apprenant-es n'ont pas les prérequis pour suivre en autonomie un cours et abandonnent alors la formation. Ils ou elles ne sont pas à l'aise avec le dispositif numérique délivrant le cours et se sentent livré-es à eux-mêmes. Ainsi, se créent des inégalités entre salarié-es confronté-es à des dispositifs de formation numériques dans leur entreprise (Boboc & Metzger, 2019a). La question des attendus impli-

cites que nécessite l'autonomie se pose donc, ici comme à l'école, bien que l'autonomie y prenne une forme tout à fait différente. Ainsi, nous souhaitons aussi dans cet article nous demander quels sont les prérequis non explicités de la formation en informatique observée. Toutefois, dans notre cas, les participant-es avaient des trajectoires et des niveaux scolaires très différents et devaient apprendre, non pas quelques compétences supplémentaires, mais un métier. Dès lors, que se passe-t-il lorsque la méthode de l'apprentissage en autonomie est la seule mise en œuvre ? Quels sont concrètement les prérequis nécessaires et en quoi accentuent-ils des inégalités entre apprenant-es ?

Encadré méthodologique

Cet article se fonde sur une observation participante de six mois dans une formation de « Data développement » à destination de personnes en recherche d'emploi en Île-de-France. La formation a duré sept mois en tout (en incluant un mois de stage en entreprise) et les apprenant-es devaient être présent-es tous les jours de 9h30 à 17h15. L'école dans laquelle a eu lieu la formation, Codon², est implantée dans toute la France ainsi que dans d'autres pays, et propose des formations courtes (de quelques mois) dans des spécialités informatiques diverses. Vingt-quatre personnes (neuf femmes, quinze hommes) étaient inscrites dans la formation au début, avec des niveaux de diplôme variés (allant du sans diplôme au Bac +5). Une dizaine de personnes avaient entre 20 et 30 ans ; la personne la plus âgée avait un peu plus de 50 ans. La formation est labellisée Grande École du Numérique (GEN), gratuite pour les apprenant-es et a été financée par les « partenaires financeurs » de la formation (dont principalement la GEN). La formation est censée permettre d'obtenir deux certifications professionnelles : un certificat AGILE (méthode de management utilisée principalement dans les start-ups) et une certification RS3497 « Développer une base de données » inscrite au Répertoire

² Les noms de l'organisme de formation et des enquêté-e-s ont été anonymisés.

Spécifique des certifications professionnelles de France Compétences et reconnue par l'État. La « *fabrique* » (nom donné aux écoles de Codon) dans laquelle a lieu la formation propose aussi un autre cursus : « Développement web ». L'apprentissage est censé être fait avec ce qui est appelé la « *pédagogie active* » où l'« *apprenant* »³ doit apprendre « *par lui-même* », jouer le rôle de l'enseignant-e pour ses co-apprenant-es, réaliser des projets basés sur des cas réels et être activement dans l'entraide. Cette idée de « *pédagogie active* » est revendiquée comme propre à l'école en question. Un-e intervenant-e est présent-e chaque jour, aux mêmes horaires que les apprenant-es. Enfin, nous avons participé à la formation en tant qu'apprenante, et nous étions perçu comme tel, tant par les autres élèves que par les intervenant-es.

De fait, certains travaux se sont intéressés à des espaces plus semblables au nôtre, c'est-à-dire dans lesquels il était question d'apprendre un métier, grâce à une pédagogie de l'autonomie censée être la seule méthode d'apprentissage. Une recherche menée par Labarthe et Delorme a porté, avec une autre perspective, sur deux formations en informatique de quelques mois censées se dérouler avec ladite pédagogie active (Labarthe & Delorme, 2019). Les apprenant-es observé-es ont tiré un sentiment positif d'avoir suivi une formation, mais ont été confronté-es à des difficultés d'insertion professionnelle. Si cette recherche partait d'un questionnement différent – en terme d'*empowerment*⁴

–, il était toutefois fait mention de difficultés liées à la pédagogie alternative. Les enseignants du dispositif de formation, issus d'établissements d'enseignement supérieur dans lesquels avaient lieu les formations, ont réintégré une pédagogie plus classique face aux difficultés éprouvées par les apprenant-es en passant par la pédagogie active. Or, peut-être parce que la formation que nous avons observée ne bénéficiait pas d'un partenariat avec un établissement d'enseignement supérieur – et donc de ses enseignant-es –, la pédagogie active a été maintenue comme cadre d'apprentissage tout au long de la formation. En effet, une pédagogie en autonomie telle que celle observée peut, *a priori*, permettre de faire l'économie de cours et des enseignant-es : en ce sens, on peut se demander si c'est pour des raisons budgétaires que la formation que nous avons observée, une formation à destination de chômeur-ses qui occupe une position dévalorisée dans l'espace de l'apprentissage en informatique, favorise autant de tels dispositifs pédagogiques.

3 Ce vocabulaire est présent tant dans les brochures de formation et livrets d'accueil que dans le discours des intervenant-e-s et autres salarié-e-s de Codon.

4 Ce mot anglais est employé tel quel dans les discours de présentation des formations enquêtées. Il est aussi analysé dans l'article de Labarthe et Delorme en s'appuyant sur les travaux de Bacqué et Biewener qui distinguent différents modèles d'*empowerment* (Bacqué & Biewener, 2015).

D'autres formations utilisent aussi une pédagogie alternative comme unique méthode d'apprentissage et vise un métier à l'issue de la période d'école : c'est le cas des formations en art. Les observations de Vandebunder, qui analyse les attendus pédagogiques en écoles d'art, présentent beaucoup de similitudes avec les nôtres. Vandebunder se pose alors la question du lien entre la pédagogie de l'autonomie et l'objectif des institutions dans lesquelles ces pédagogies s'inscrivent : transformer des individus en artistes. En école d'art, l'autonomie est ainsi perçue comme un signe de statut artistique : l'étudiant doit donc trouver le juste milieu entre être artiste et rester étudiant. Vandebunder conclut en relevant que le passage de l'étudiant à l'artiste relève autant de la pédagogie que de la magie, à l'instar des brahmanes analysés par Weber, puisqu'il s'agit de révéler des caractéristiques que les étudiant-es posséderaient préalablement (Vandebunder, 2015). Nous souhaitons aussi nous poser la question de la relation entre les formes pédagogiques observées et l'objectif de la formation en informatique : ainsi, dans quelle mesure cette pédagogie permet-elle aux apprenant-es de devenir développeur-ses ? Cependant, contrairement aux écoles d'art de Vandebunder, la formation en programmation observée, dans une école qu'on appellera Codon, est une formation de quelques mois à destination de chômeur-ses : en ce sens, elle n'est pas comparable aux écoles d'art qui délivrent des diplômes de niveau bac +5. Par sa durée (relativement courte), par son public (dans une situation de

précarité, d'une part, mais aussi caractérisé par sa diversité en termes de niveau de diplôme et d'âge notamment), et par sa visée (d'inclusion sociale et d'insertion professionnelle), elle s'éloigne des formations en informatique que sont les écoles d'ingénieurs, les écoles d'informatique comme 42⁵, SupInfo, Epitech, et les facultés d'informatique. On peut supposer qu'à Codon plus qu'ailleurs, il faudrait soit une pédagogie très efficace – pour qu'elle s'adapte à des profils très différents et les transforme en travailleurs recherchés par les entreprises pour leur compétence technique rare⁶ –, soit beaucoup de « magie » – pour reprendre le terme employé par Vandebunder. Or, peut-on se reposer sur la « magie »,

⁵ Si 42 et Codon présentent certains points communs (notamment un discours d'inclusion sociale), le parcours de formation à 42 est généralement plus long (plusieurs années) que celui chez Codon. De plus, à 42, les étudiant-es sont sélectionné-es à l'issue de plusieurs épreuves de programmation – ce qui n'est pas le cas chez Codon. La population des étudiant-es de 42 apparaît donc sensiblement différente de celle des formations courtes. Par ailleurs, 42 cherche à se distinguer des formations de quelques mois. Voir par exemple sur le site de 42 [URL : <https://42.fr/42-cest-quoi/la-methode-42/>] : « À l'opposé des formations courtes, spécialisées dans un domaine ou un langage qui peut devenir obsolète en quelques mois, 42 permet d'affronter avec sérénité les changements technologiques tout au long de sa carrière. La méthode 42 repose sur "apprendre à apprendre". C'est la capacité à s'adapter et à trouver des solutions concrètes à des problèmes posés en se basant sur les connaissances et outils actuels. »). Pour en savoir plus, voir l'ouvrage de Dupuy et Sarfati (Dupuy, Sarfati, 2022).

⁶ Les formations en informatique labellisées Grande École du Numérique (GEN), comme Codon, parlent d'une pénurie de développeurs-ses sur le marché du travail. Cette idée de pénurie est toutefois à déconstruire en tenant compte qu'elle est le fruit d'un discours patronal (Zune, 2006).

comme dans les écoles d'art étudiées par Vandenburg, quand il s'agit de personnes en recherche d'emploi ?

On proposera d'abord une description de ce cadre d'apprentissage : comment il a été présenté aux apprenant-es, mis en œuvre et adopté par les acteurs-rices de la classe observée. Ce sont peut-être les modalités d'enquête – une observation participante à couvert – qui nous feront dans un second temps approcher l'objet en partant du point de vue du dit apprenant : comment apprend-il dans ce cadre d'apprentissage ? On verra comment le travail traditionnellement laissé à l'enseignant lui incombe, favorisant dès lors les apprenant-es les plus aptes à enseigner.

La réception de la pédagogie active : entre adhésion et difficultés d'adaptation

La présentation de la pédagogie : un dispositif d'apprentissage justifié par de nombreux éléments

Codon organise régulièrement des séances de présentation de ses formations à Pôle Emploi. Ces réunions semblent avoir beaucoup de succès : la salle de celle à laquelle nous avons assisté était comble et la réunion précédente avait aussi attiré trop de monde par rapport à ce qui était prévu. L'organisme de formation se distingue d'autres formations en informatique par son dis-

cours centré sur la valorisation de l'aspect « social et solidaire ». Ainsi, lors de cette réunion, les deux intervenant-es de Codon parlent d'un « *but éthique* », d'une visée « *sociale* » ; de transformer un secteur réservé à « *des hommes diplômés* » en l'ouvrant à des femmes et à des personnes non diplômées. La réunion est l'occasion de présenter des statistiques produites par Codon : 51 % des personnes en formation ont un niveau bac ou infra-bac, 31 % sont des femmes. Codon se présente aussi, notamment dans sa communication en ligne via son site Internet, comme un organisme différent parce qu'il propose une pédagogie « *innovante* », la « *pédagogie active* », qui est présentée comme une fabrication originale de Codon, et qui s'adapte à tous les profils⁷ – très différents donc – et les adapte à l'entreprise.

À Codon, dès le processus de recrutement, le candidat est confronté à l'apprentissage en autonomie. Le recrutement des apprenant-es se passe en deux étapes. La première consiste à remplir un formulaire en ligne, accessible par toute personne via le site de Codon, avec des questions sur le parcours du-de la candidat-e ainsi que sur ses motivations. La deuxième consiste à passer une demi-journée dans un lieu de formation. Prévenus une semaine à l'avance, les candidat-es ont durant cette demi-journée trois entretiens individuels de huit

⁷ Cette idée d'adaptation à chacun-e est reprise par certain-e-s intervenant-e-s pour justifier la forme pédagogique de la formation.

minutes ainsi qu'un « *projet* » à réaliser en groupe. Dans la demi-journée à laquelle nous avons participé, il s'agit de faire un croquis de « *front page* » (la devanture d'un site web). Nous sommes incité-es à faire des groupes mixtes dans la mesure du possible (seules trois femmes sont parmi les candidat-es lors de cette demi-journée), ce qui échoue assez rapidement puisqu'un groupe de trois est formé avec deux femmes et deux groupes sont composés exclusivement d'hommes. À la fin de la demi-journée, les projets réalisés sont présentés à l'ensemble des candidat-es et des recruteur-ses. Les candidat-es recruté-es l'apprennent quelques jours plus tard : c'est le cas des deux autres personnes de notre groupe de demi-journée. N'ayant pas eu accès aux discussions des recruteur-ses, ni à des statistiques sur les candidat-es, il est difficile de dire quels critères sont utilisés en pratique pour le recrutement ; toutefois, on peut noter que la GEN labellise (et donc finance) avec des critères qui concernent les personnes en formation (mixité femmes / hommes (objectif de 30 % de femmes) ; niveau de diplôme (objectif de 80 % de personnes ayant un niveau de diplôme équivalent bac ou inférieur)). De plus, Codon communique sur la diversité de ses publics.

La première semaine de la formation est consacrée à poser un cadre d'apprentissage et en particulier à expliquer et à valoriser la pédagogie proposée par l'école. Le responsable du lieu de formation a, dès le premier jour, envoyé plusieurs documents : la « Charte

d'engagement »⁸ à destination des apprenant-es, le « Règlement intérieur des apprenant-es » et le programme de formation. C'est ce dernier qui explicite les prérequis ainsi que la « *pédagogie active* ». Parmi les prérequis attendus, on peut lire « *autonomie* », « *anglais parlé et lu* », « *patience, persévérance, détermination, débrouillardise, rigueur, curiosité* », ainsi que « *motivation éprouvée et quantifiée par des épreuves d'autoapprentissage préalables* ». Si ces prérequis sont flous, il est clair que pour l'anglais, les apprenant-es observé-es n'ont pas les prérequis à l'entrée en formation. En effet, au moins la moitié de la classe ne maîtrise pas, ou très peu, cette langue.

Quant à la « *motivation éprouvée et quantifiée par des épreuves d'autoapprentissage préalables* », elle fait référence à un attendu de l'admission en formation : chaque apprenant doit en effet avoir réalisé des « *badges* » sur une plateforme d'enseignement du code informatique proposée par une start-up américaine. Ces badges, qu'on obtient en faisant des exercices à partir d'une courte leçon, sont en partie gratuits. Ainsi, à l'arrivée, chacun-e est censé-e avoir réalisé intégralement les cours, gratuits, de Python⁹

⁸ La charte d'engagement stipule que l'apprenant-e s'engage à « *contribuer à la réussite collective* », « *s'impliquer dans les projets proposés* », « *chercher activement un emploi* », et « *respecter le cadre proposé* ».

⁹ Python est le langage informatique enseigné et utilisé ensuite dans la formation. C'est un langage connu pour sa simplicité. Populaire, il est en particulier beaucoup utilisé dans le secteur de « la data », au côté de R.

(version ancienne) et de SQL¹⁰ proposés sur la plateforme. L'apprentissage en solitaire a donc commencé préalablement à la formation. Dès les épreuves d'admission, la formation n'est rendue possible que par l'existence de contenus d'enseignement et d'apprentissage créés par d'autres (ici, une entreprise californienne) et utilisés gratuitement sur demande de l'organisme de formation.

L'une des premières après-midi est consacrée à expliquer la pédagogie mise en œuvre dans la formation. L'intervenant présent explique, au tableau : « *Étudiant = acteur = faire / expérimenter / chercher* ». Selon lui, « *chacun a sa sensibilité* », et il s'agit donc d'avoir une « *pédagogie adaptée à chacun* ». Bien que la pédagogie soit présentée comme propre à Codon, on y retrouve en fait plusieurs éléments pédagogiques – comme la notion d'« apprenant », l'« autonomie » et le fait que l'apprenant est censé construire ses savoirs – qui sont aussi présents dans d'autres formations numérisées (Boboc & Metzger, 2016 ; Boboc & Metzger, 2019b).

Le responsable de la formation vient aussi présenter les types de public en formation dans toute l'école et spécifiquement dans cette classe : c'est l'occasion de montrer que l'entreprise remplit bien ses objectifs de féminisation des formations informatiques, de diversité des niveaux de diplôme, d'âge, et d'inclusivité des per-

sonnes en situation de handicap. Dans la présentation, revient de plus beaucoup l'idée d'entraide, puisque chaque apprenant-e doit apprendre à l'autre, expliquer si l'autre en a besoin, demander s'il ne sait pas. Le principe d'entraide est par la suite très intégré dans le discours des apprenant-es. Lorsque le jour suivant a lieu un atelier « Règlement intérieur » dans lequel chaque participant doit, à tour de rôle, venir mettre un post-it sur le tableau, « *s'entraider* » revient sur plus de 70 % des post-it.

Le discours présentant la pédagogie active est efficace : la grande majorité des apprenant-es valoriseront ce dispositif d'apprentissage jusqu'à la fin de la formation. En effet ce dispositif s'appuie sur différentes justifications.

Premièrement, comme on l'a évoqué en introduction, le fait que l'informatique nécessiterait un apprentissage particulier. Ainsi, bien que Codon revendique une pédagogie originale, plusieurs écoles d'informatique, privées, mettent exclusivement en œuvre ce type de pédagogie : 42, l'école emblématique de Xavier Niel, mais aussi Epitech, l'école censée être le modèle de 42, et SupInfo par exemple. Puisque l'informatique est un milieu dans lequel « les technologies évoluent perpétuellement »¹¹, il faudrait que tout-e professionnel-le soit en mesure de s'adapter et d'apprendre soi-même les nouveaux outils auxquels il aura à faire face¹².

¹⁰ SQL est un langage qui permet de faire de la gestion de bases de données. Il est enseigné dans de nombreuses formations initiales d'informatique.

¹¹ Cette idée est un lieu commun chez les informaticien-nes.

¹² Voir la note de bas de page supra sur 42.

Deuxièmement, dans ces écoles, il s'agit aussi de ne pas laisser de côté des « génies de l'informatique » que l'école n'aurait pas su révéler. Le milieu de l'informatique est en effet marqué par la figure de l'informaticien-hacker : un homme passionné par la programmation, peu sociable, se moquant de la réussite professionnelle (Collet, 2006). C'est celui-ci, qui code la nuit, mais qui pourrait ne pas avoir révélé son potentiel dans le système scolaire classique, à qui il faudrait un apprentissage adapté (horaires souples, pédagogie en opposition à une pédagogie traditionnelle). En particulier, la Grande École du Numérique se présente comme une école de la « seconde chance ».

Troisièmement, le fait que cette pédagogie permettrait à chacun-e d'avancer à son rythme : ainsi, la formation est individualisée pour que chacun-e progresse à sa vitesse sans être ralenti-e par le groupe ou au contraire, décrocher, par manque de prérequis. Cette individualisation de la pédagogie s'inscrit dans un discours qui met en avant la diversité des apprenant-es (personnes en situation de handicap, personnes avec différents niveaux de diplômes, personnes d'âge et de sexe différents, personnes ne parlant pas français). Ainsi, la pédagogie active serait adaptée à tous les profils.

Enfin, le fait que la pédagogie active serait une pédagogie plus efficace, notamment en vue de l'objectif de la formation : être adapté-e aux attentes des entreprises pour obtenir et conserver un emploi. En entreprise, le ou la développeur-se devrait en

effet être autonome et en mesure de réaliser des « projets » – pour reprendre la rhétorique codon – tout de suite. Or, la pédagogie active permettrait d'appliquer, de favoriser la pratique et de se confronter à des projets concrets. Cet aspect constitue donc un autre principe d'adhésion des apprenant-es à la pédagogie active : pratiquer et reproduire ce qui se passe en entreprise leur apparaît comme la voie la plus sûre pour trouver un emploi. À cet égard, on peut observer de nombreux points communs avec les apprentis en collègue d'enseignement technique qui sont l'objet de l'ouvrage de Claude Grignon, *L'ordre des choses*, et qui valorisent la pratique plus que la théorie, un travail manuel, appris par l'expérience, en opposition à une culture savante, futile, qui s'acquiert de façon passive en écoutant un discours plutôt qu'en faisant (Grignon, 1971). Le dispositif pédagogique observé s'appuie donc sur un esprit anti-scolaire, sur lequel s'adosent aussi, historiquement, les formations pour adultes, ainsi que, comme le remarque Michaël Vicente, le reste des formations courtes en informatique (Vicente, 2019).

**« J'hésite à me pointer cet aprem »¹³ :
l'organisation de l'apprentissage
avec la pédagogie active**

Dans les faits, le cadre d'apprentissage va beaucoup évoluer au cours du temps, en fonction des intervenant-es et du programme supposé de la formation.

¹³ Message envoyé par un apprenant à d'autres apprenant-es via le forum de discussion en ligne.

Tout se passe dans une salle de classe attribuée à chacun des deux groupes ainsi que dans l'espace commun aux deux groupes (comportant une cuisine et quelques canapés). Chaque apprenant-e utilise un ordinateur portable (le sien ou celui prêté par l'école).

Durant les premières semaines de la formation, la journée est assez organisée : le matin, il faut faire une veille, c'est-à-dire une recherche sur un sujet donné par la formatrice ou le formateur, dont on doit rendre compte à 11h¹⁴. Ensuite, la plupart du temps, des travaux pratiques (« TP ») sont envoyés sur l'application de discussion et d'échanges de la classe. Il s'agit de faire des exercices, seul-e, sur « *une technologie* »¹⁵. Mais dès le deuxième mois, l'organisation de l'apprentissage change puisqu'il s'agit désormais de faire un premier projet. La classe va aussi avoir plusieurs jours consacrés à l'enseignement de la méthode Agile, et les enseignant-es vont changer. Enfin, les trois mois après le mois de stage en milieu professionnel seront principalement consacrés à la réalisation d'un « *chef-d'œuvre* », projet présenté en fin de formation devant un jury afin d'obtenir une certification. Ainsi de nombreuses journées (trois jours par semaine en moyenne, parfois plus, parfois moins) y sont consacrées :

chaque apprenant-e se retrouve en autonomie du matin au soir et doit chercher des idées sur ce qu'il/elle pourrait faire puis réaliser ce projet.

Il n'y a pas de cours classique : l'enseignant-e n'est jamais debout au tableau, ou face aux élèves ; ses rares prises de parole sont la plupart du temps pour dire qu'il/elle a envoyé un mail avec les consignes. Il n'y a pas non plus de cours théorique : un concept informatique peut apparaître en fin de TP mais seulement sous la forme d'un intitulé d'exercice et du détail de celui-ci. Pour faire les exercices, il est la plupart du temps nécessaire de trouver et de lire un cours ou un tutoriel préalablement. Par exemple, parmi les cours les plus utilisés se trouvent le cours de Python proposé par l'Université Diderot en version .pdf¹⁶, et les cours issus d'un site de Moocs (cours en ligne) français. Les apprenant-es qui ont un niveau d'anglais suffisant vont aussi sur des plateformes anglophones de Moocs. La formation n'est donc possible que parce que d'autres personnes ont créé du contenu d'apprentissage et l'ont rendu disponible. À cela, il faut ajouter les épreuves d'admission qui elles-mêmes s'appuient sur la version gratuite d'une plateforme d'apprentissage en ligne. Enfin, les encadrant-es eux-mêmes proposent, en guise de TP ou de projets, des contenus qu'on retrouve en ligne, issus d'école d'ingénieur, ou d'autres lieux de formation. L'école observée entre-

¹⁴ Les sujets peuvent varier : « Qu'est-ce qu'une librairie Python ? », « Qu'est-ce qu'un bon code ? », « Qu'est-ce qu'un algorithme ? », « Qu'est-ce qu'un tuple ? ».

¹⁵ Cette technologie peut être un langage informatique (Python par exemple) ou un module (Flask par exemple).

¹⁶ Format de document numérique répandu.

tient donc une dépendance très forte à l'égard d'autres institutions (université, écoles d'ingénieurs) ou d'autres créateurs de contenu.

Le premier intervenant de la formation conseille aux apprenant-es de prendre l'habitude à chaque blocage de chercher sur Internet. On incite systématiquement l'apprenant-e à « *chercher par soi-même* » avant de poser la question à quelqu'un d'autre, ou *a fortiori* à l'enseignant : il s'agit alors de poser toute question d'abord au moteur de recherche. Chaque apprenant-e doit d'abord essayer de se débrouiller seul avant de demander de l'aide, ce qu'on peut lire comme une injonction à la responsabilisation des individus. Si l'apprenant-e n'y arrive pas et qu'il demande à l'enseignant-e, il lui est suggéré de demander aux autres apprenant-es d'abord. Les apprenant-es qui par la suite se plaignent de ne pas y arriver sont à nouveau incité-es à chercher activement de l'aide en se levant pour aller demander aux autres ; à l'inverse, celles et ceux qui ont réussi doivent dire explicitement qu'ils sont disponibles sur l'application de discussion de la classe. Certes, comme l'écrivait Lahire, on peut voir ceci comme une « *dépersonnalisation du savoir* » (Lahire, 2001, p. 158). Mais, après tout, dans l'esprit des apprenant-es, quelqu'un est censé mieux savoir que les autres : l'enseignant-e. Ainsi, on apprend à ne pas demander à la personne qui a le plus de ressources, mais d'abord aux personnes qui sont nos égaux.

Cette méthode d'apprentissage est critiquée par certain-es apprenant-es :

« C'est facile d'être prof ici. Quand tu ne sais pas, tu dis : cherchez sur Google, ou mieux : quelqu'un a la réponse ? »

Anas, 29 ans

Mais puisque chacun doit trouver la leçon pour l'apprendre et essayer de l'apprendre seul, le temps est aussi géré en autonomie. Ainsi, il est normal que chacun-e fasse ce qu'il veut au moment où il le veut : nul besoin de se conformer à des horaires ou de faire la même chose que le reste du groupe.

Extrait de journal de terrain (31 juillet) :

La chaise d'un apprenant couine : il ne s'en rend pas compte car il a un casque sur les oreilles. Le prof, qui parle à l'ensemble du groupe, finit par s'arrêter et le regarde comme tout le monde. Le prof lui fait un sourire et l'homme enlève son casque.

Cet extrait de journal de terrain n'est pas sans rappeler les constats que faisait Jérémie Vandebunder, évoqué précédemment, en observant les écoles d'art dans lesquelles les étudiants entraient et sortaient dans les salles de classe au gré de leurs envies (Vandebunder, 2015). Une partie des apprenant-es – principalement des hommes – ne parvient pas à travailler : les midis sont souvent l'occasion de se raconter qu'on n'a rien fait le matin. C'est aussi le cas des pauses qui s'étendent pendant parfois une heure.

Mais d'autres apprenant-es – principalement des femmes – se plaignent d'être très fatigué-es, de travailler la journée et le soir, et d'avoir des difficultés à réaliser les projets. On constate ainsi une différence sexuée dans les réactions aux injonctions à l'autonomie, qui peut être lue au prisme des recherches sur le genre à l'école (Mosconi, 1999).

On peut supposer que cette façon de mettre en application l'autonomie a favorisé les émotions négatives observées sur le terrain. En effet, les apprenant-es se plaignent à plusieurs reprises de ne rien comprendre, d'être « *perdu-es* », de ne pas s'en sortir. À cet égard, les femmes ont, *a priori*, exprimé plus d'émotions négatives, sans qu'il soit possible de déterminer si c'est parce qu'elles ont ressenti plus de sentiments négatifs que les hommes, ou parce qu'elles les auraient davantage exprimés – les deux hypothèses étant plausibles considérant les recherches actuelles (Simon, 2014). Or, les recherches en sciences de l'éducation montrent aussi que les émotions dites positives favorisent l'apprentissage, et que les émotions dites négatives diminuent les performances d'apprentissage (Govaerts & Grégoire, 2006). Ces émotions observées entrent aussi en contradiction avec les objectifs du code intégré (Bernstein, 1975), qui met l'accent sur la manière d'apprendre et sur l'épanouissement de l'apprenant-e dans ce processus d'apprentissage. On peut ainsi faire le constat d'un décalage entre un discours qui vante une pédagogie qui serait plus efficace et le vécu des apprenant-es.

Omniprésence de l'appellation « projet » : de l'exercice à la construction d'un « chef-d'œuvre »

La pédagogie active met le projet au centre de l'apprentissage : de nombreuses semaines y sont consacrées. Il a été demandé aux apprenant-es de réaliser plus d'une quinzaine de projets. À cela s'ajoutent le projet individuel de fin de formation ainsi qu'un autre projet long. La notion de projet est donc omniprésente, ce qui permet de souligner l'imprégnation du nouvel esprit du capitalisme dans la formation (Boltanski & Chiapello, 1999).

La pédagogie par projet impliquant de faire faire aux apprenant-es des projets, tout exercice semble pouvoir être dénommé « projet ». Par exemple, il est demandé aux apprenant-es de reproduire une base de données SQL¹⁷ simulant une base de données commerciale : il s'agit alors d'écrire une instruction¹⁸ pour créer une table « clients » et une table « commandes »¹⁹. Le projet ne consiste pas à

17 Pour une définition de SQL, voir *supra*.

18 Pour avoir une idée de la quantité de travail à fournir, une instruction (ou « commande », dans le jargon informatique) en langage SQL pour créer une table consiste à écrire « CREATE TABLE » suivi du nom de la table (par exemple « Clients »), du nom des colonnes (par exemple « Nom », « Adresse », « Téléphone ») et du type de données (« VARCHAR(100) » pour texte court). L'ironie étant que les apprenant-es sont incité-es à regarder sur Internet et que c'est un des premiers exemples montrés sur la plupart des sites web.

19 Une table correspond à un tableau. Dans l'exemple mentionné, ce sont des tableaux listant les « clients »

se demander comment créer la base de données d'une entreprise (impliquant de réfléchir, par exemple en choisissant le type de base de données et en se demandant comment l'organiser), mais à trouver les lignes de commande SQL pour en reproduire une déjà construite. En ce sens, c'est un exercice pratique, qui consiste à appliquer quelques instructions apprises, en éliminant tous les aspects réflexifs.

Un autre projet consiste à récupérer une base de données issues du site Airbnb²⁰ à propos des types d'appartements loués à Paris et à « *les analyser* », sans consigne particulière autre qu'essayer de faire de la « *data visualisation* », c'est-à-dire des graphiques, à partir de ce jeu de données. Il s'agit alors de « *s'inspirer* » de ce qui est fait sur un site consacré à l'analyse de données (Kaggle) où la base de données a pu être récupérée. Le projet est censé durer quatre jours. Certains projets ont pour seule consigne d'utiliser un outil. Les projets – parfois individuels, parfois en groupe – doivent être présentés devant le reste de la classe mais il est toujours possible d'être absent-e à ce moment-là ou de refuser de le présenter. La présentation du projet consiste à montrer le code et les résultats.

Trois projets longs sont à réaliser durant la formation : le projet individuel de fin de formation, le projet « outil de

conversation instantanée » et le projet FGH²¹. Dans ce dernier projet, il s'agissait d'aider une association réelle à informatiser une partie de son travail.

Extrait de journal de terrain (28 octobre) :

Le projet FGH est un projet confié à l'école Cergy de Codon par une association qui aide les étrangers à obtenir un titre de séjour ou un regroupement familial. En quelques jours, les consignes, données par les encadrant-es, ont été changées au moins trois fois. Les encadrant-es ne semblent pas s'être coordonné-es sur les consignes à donner. Ils paraissent ne pas savoir comment réaliser le projet eux-mêmes.

Nous y avons passé environ une semaine avec un intérêt informatique pour le moins très limité : il s'agissait de faire un fichier json (à l'occasion, nous avons appris à quoi ça ressemblait, mais sans savoir à quoi ça sert) que nous avons réalisé à coup de copiers-collers. Seul-es certain-es avaient compris comment organiser le document json et ils expliquaient donc aux autres ce qu'ils devaient copier-coller, pour qu'à la suite, tout soit assemblé manuellement. De fait, le fichier que nous avons fait n'était pas valide : en tout cas, il n'a pas fonctionné. Aucun des quatre encadrant-es ne savait comment faire au bout de plusieurs jours. Omar est le seul à avoir vraiment travaillé dessus – la plupart du groupe a abandonné rapidement – et à avoir réussi à faire quelque chose

et leurs « commandes » (à ne pas confondre avec la « commande » au sens de l'instruction).

²⁰ Site web et application de locations de vacances gérées par les utilisateurs.

²¹ Le nom original du projet vient de l'acronyme de l'association à l'origine de ce projet.

de fonctionnel, qu'il dit basique mais que personne ne sait faire. Sirine, une autre apprenante, a dit que ça aurait dû être un projet proposé à des volontaires et non à l'ensemble du groupe. Selon elle, à vingt, personne n'ose prendre les choses en responsabilité et les faire complètement. Certain-es se trouvent une tâche et la font, mais la plupart ne font rien.

L'ensemble des projets qu'on nous a demandé de réaliser ont été laborieux, critiqués par tou-ttes les apprenant-es, et anxigènes pour de nombreuses personnes. Les consignes n'y étaient pas claires et, selon le formateur présent, pouvaient changer. Aussi, plusieurs projets sont marqués par le manque de connaissances des apprenant-es afin de les réaliser.

Extrait de journal de terrain (15 août) :

Nous ne sommes que quinze aujourd'hui, vendredi. Beaucoup d'absent-es. Le prof présente quelque chose jusqu'à 10h30 puis nous laisse sur un TP « élections présidentielles ». Il a aussi, comme les fois précédentes, noté les consignes sur Discord :

« Programme de la semaine prochaine :

Lundi et Mardi : Avancer sur le projet élections présidentielle.

Mercredi : Correction et restitution du Projet

Jeudi et Vendredi : Avancer sur le Projet Grande Distribution.

Merci. »

La pédagogie active : un dispositif qui laisse une large marge de manœuvre à l'enseignant-e

Si en apparence la pédagogie active peut permettre, sinon de se passer de professeur-e, au moins de lui donner un rôle moins important – puisqu'il ou elle est censé-e être un-e accompagnateur-riche –, cela se révèle en pratique très peu le cas. En fait, la dépendance à l'encadrant-e peut même apparaître accrue : selon l'accompagnateur-riche avec lequel le groupe se trouve, l'apprentissage est extrêmement différent. Le cadre d'apprentissage suggéré par la pédagogie active laisse énormément de marges de manœuvre et de nombreuses possibilités d'application : du terme « projet », flou, à ce que doit faire l'apprenant-e pour être « actif » et « autonome ». Sur ce dernier point, par exemple, l'apprenant-e peut être considéré-e comme actif-ve soit, lorsqu'à partir d'une consigne journalière, il se débrouille seul toute la journée (sans avoir besoin de consignes intermédiaires) et réalise, sans intervention de l'encadrant-e, un travail entrant dans le cadre de ce qui est demandé ; soit, lorsqu'il fait effectivement quelque chose pouvant être considéré comme de l'informatique (c'est-à-dire qui ne soit pas regarder une série par exemple), mais sans lien avec le projet en cours. Si certains encadrant-es interviennent auprès de certain-es apprenant-es individuellement, ce qui peut être perçu à la fois comme une proposition d'aide et comme un moyen de contrôle, d'autres ne le font pas, considérant que l'autonomie implique que les apprenant-es soient seul-es.

La pédagogie active, par la souplesse des termes qui composent ses principes (« autonome », « actif », « projet »), laisse place à de nombreuses possibilités dans sa mise en pratique. Ces observations vont dans le sens d'autres recherches menées en maternelle, qui montrent que l'application d'une pédagogie alternative peut se révéler très éloignée de ce qu'elle est censée être. En l'occurrence, ces recherches établissent que dans la pratique, les enseignantes continuent à fabriquer un cadre d'apprentissage classique, avec des rôles traditionnels maître / élève, et une utilisation des projets comme dispositif d'apprentissage très limitée, même pour les personnes qui se revendiquent partisans des pédagogies alternatives (Leroy, 2017). Chez Codon, les mises en pratique de la pédagogie active se révèlent très différentes selon l'intervenant-e.

L'un des intervenants, Alexandre, est très vivement demandé, mais l'entreprise de formation préfère le placer sur des formations financées par des entreprises, de sorte qu'il ne viendra au total que quelques jours encadrer la formation observée. Ingénieur de formation, ayant complété cette formation initiale par d'autres masters, Alexandre est *freelance* (à son compte). Il est le seul intervenant à dire qu'il enseigne à Codon même s'il est moins bien payé qu'ailleurs parce qu'il apprécie les valeurs de l'école. Alexandre parvient à mettre en place des projets qui sont adaptés en termes de niveau pour chacun-e et explicite aussi très rapidement ce qui peut être intéressant à faire pour les apprenant-es dans les projets commencés

(n'hésitant pas à critiquer ce qui n'a à ses yeux pas d'intérêt dans les exercices lancés par d'autres intervenant-es). Il peut de plus proposer de diviser un projet long en étapes afin de guider les apprenant-es sur l'avancement. Il parvient à distribuer son temps entre chaque apprenant-e : il vient vers chacun-e – ceux qui demandent comme ceux qui ne demandent pas – et personne ne se plaint d'avoir dû passer un temps trop long avec lui – ce qui est le cas pour d'autres enseignant-es. Il invite aussi à essayer d'apprendre certaines notions afin que chacun puisse avancer sur le projet. Après avoir laissé le temps aux apprenant-es de chercher les savoirs, il demande au groupe de lui expliquer, et fait ensuite une reprise explicative.

Extrait de journal de terrain (30 octobre) :

Alexandre le prof est là (après une absence de plusieurs mois). Il propose sept ateliers différents parmi lesquels nous pouvons choisir. Je me mets à la première table (atelier scraping et NLP). Et nous commençons à travailler chacun-e de son côté. À la différence des autres profs, Alexandre vient vers nous au moins une fois par heure demander où on en est et donner des indications claires sur ce qu'il faut faire, expliquer les éléments conceptuels (par exemple qu'est-ce que TL-IDF ?) en faisant des schémas au tableau.

Il apparaît qu'être un-e encadrant-e en pédagogie active nécessite beaucoup de compétences ainsi qu'un temps complètement dévolu au groupe. Étant extrêmement large – *a priori* trop selon

certains professionnels interrogés –, le programme demande par conséquent des connaissances informatiques extrêmement variées. Or, tous les encadrant-es ne disposent pas également de ces compétences et ne semblent pas avoir non plus le temps sur leur temps de travail pour s’y consacrer pleinement. Ce constat fait écho à la conclusion de la recherche de Lahire en école primaire, qui soulignait l’aspect chronophage de la mise en place de l’autonomie à l’école, puisqu’elle demande un long de travail de préparation aux enseignant-es ainsi qu’un ensemble de ressources à proposer aux élèves (Lahire, 2001).

Un autre intervenant, Julien, fait l’objet de fortes critiques de la part du groupe. Au contraire d’Alexandre, Julien n’a aucun discours qui témoignerait d’une adhésion aux valeurs de Codon. Il apparaît aussi avoir une position moins privilégiée qu’Alexandre sur le marché de l’emploi. Les critiques émises à son égard lors d’une réunion entre délégué-es de classe et responsables pédagogiques se fondent sur la défense de la pédagogie active : il est reproché à l’enseignant de ne pas assez la mettre en œuvre en proposant des sessions de formation dans lesquelles les apprenant-es sont passif-ves tandis que lui projette son écran d’ordinateur au tableau et commente. Cette critique n’est pas partagée par l’ensemble du groupe : certaines apprenantes trouvent justement que « *c’est bien qu’il prenne le temps de tout expliquer, de tout montrer, de repartir du début* ». Lors d’une présentation d’un

projet de fin de formation réalisé dans une autre formation, une ancienne apprenante explique qu’il y a eu des « *problèmes* » dans sa formation, parce que les apprenant-es auraient été laissé-es cinq mois avec ce même enseignant, avec lequel ils auraient peu appris. La critique de l’ancienne apprenante porte notamment sur le fait que les apprenant-es étaient livré-es à eux-mêmes et ne pouvaient espérer que très peu d’aide ou d’explication de l’enseignant (il lui est reproché de renvoyer les apprenant-es au moteur de recherche Google). Plus tard, l’enseignant est défendu parce que c’est aux apprenant-es de se débrouiller, d’apprendre par eux-mêmes. L’opposition à l’intervenant ne peut donc pas passer par une critique de la pédagogie active.

Extrait de journal de terrain (26 août) :

Retour de Lela, la formatrice principale. Debrief avec elle de ces deux mois. Les plaintes sont nombreuses. Petit débat autour de Julien. Lela reprend : « en fait chez Codon un formateur, c’est pas un prof, c’est un accompagnateur. Il accompagne les élèves à apprendre tout seul. »

Il n’en reste pas moins que le rôle dévolu à l’encadrant peut présenter une ambiguïté : il faut qu’il aide l’apprenant-e mais pas trop. Ce discours malléable sur la pédagogie permet de se défendre face à des apprenant-es : l’enseignement donné ne peut pas ne pas être adapté, puisque c’est à chacun de le concevoir, en autonomie, et mieux vaut n’avoir personne pour leur apprendre l’informatique puisqu’il

faut apprendre à apprendre justement. Cette conception est peut-être d'autant plus utilisée qu'elle peut profiter à l'entreprise de formation. En effet, de fortes difficultés à trouver des formateur·rices sont soulignées tout au long de la formation par le responsable de l'école. Il est notamment arrivé qu'un encadrant ait été appelé pour le matin même du jour où il doit venir (l'encadrant était alors arrivé en retard).

La question se pose alors de la pédagogie active comme outil de la formation *low-cost* (à bas coût) : en entreprise, les formations numérisées reposant sur l'autonomie des salarié·es « apprenant·es » permettent de réduire les coûts (Benedetto-Meyer, Boboc & Metzger, 2019). À Codon, ces coûts de formation sont amoindris par le fait qu'il n'y a pas besoin de payer d'enseignant·es, de personnes qui créent un cours, puisqu'un « accompagnateur » suffit. De plus, la rémunération des intervenant·es est, chez Codon, plus faible qu'ailleurs – on comprend alors aisément qu'il soit particulièrement difficile de trouver des formateur·rices.

Comment l'apprenant·e apprend ?

Délégation du travail d'enseignement à l'élève

De nombreux savoirs implicites sont nécessaires à ce qu'on appellera ici l'autoapprentissage.

Chercher, trier, sélectionner les savoirs

La première étape de l'apprentissage consiste à trouver une ressource qui va permettre d'obtenir les connaissances à apprendre : un cours en ligne, un article de blog, un tutoriel. Durant l'enquête de terrain, nous avons pu constater que cette étape nécessite beaucoup de prérequis, prérequis qui peuvent donc pénaliser dès le départ les apprenant·es les moins dotées, accentuant alors leur retard sur l'ensemble du groupe.

Extrait de journal de terrain (2 juillet) :

Ma camarade de projet confond les pubs et les résultats de recherche du moteur de recherche. C'est aussi le cas sur les sites internet qu'elle visite où elle clique indifféremment sur des publicités ou des hyperliens proposés par le site. Lorsqu'elle est en train de faire une recherche en ligne, elle ouvre des sites qui n'ont rien à voir avec sa recherche. Par exemple, des sites qui indiquent « nourriture pour chat » alors que son objectif de recherche est de savoir comment créer un « chat »²² en ligne. Sa recherche est d'abord faite en anglais « chat box » puis ensuite faite en français « boîte chat », ce qui continue à donner des résultats principalement liés aux animaux.

Cet extrait de journal de terrain illustre le premier temps de la recherche de connaissances. Pour faire une recherche en ligne, il faut en effet choisir ses mots,

²² Une conversation en ligne.

ce qui nécessite bien entendu des dispositions linguistiques – un rapport à la langue écrite – dont chacun-e est inégalement doté-e. Il s’agit à la fois d’avoir le vocabulaire dans la langue adaptée mais aussi de connaître la façon d’exprimer sa recherche à un moteur de recherche. Par exemple, par habitude, la plupart des utilisateurs tendent à ne chercher que quelques mots clés, sans faire de phrases, et à éviter les mots équivoques. De plus, que l’on ait choisi un premier ensemble de mots clés pertinents ou non, il faut ensuite savoir décortiquer les résultats de recherche afin de savoir si les mots clés utilisés convenaient. Ceci nécessite une capacité à lire rapidement aux bons endroits de la page Internet pour savoir si la recherche a été efficace ou non.

Certaines enquêtes ont montré que les personnes peu dotées en capital culturel faisaient face à de nombreuses difficultés lorsqu’il s’agissait d’explorer Internet (Granjon, Lelong & Metzger, 2009). Depuis, comme le remarque Dominique Pasquier, les classes populaires sont désormais quasiment aussi connectées que les autres fractions sociales. Cette dernière souligne aussi les capacités d’exploration et l’apprentissage en ligne fait par les familles modestes : ainsi, la plupart des enquêté-es de Dominique Pasquier trouvent ce qu’ils cherchent. L’apprentissage en ligne des personnes de classe populaire via la lecture ou via des tutoriels est aussi très courant (Pasquier, 2019). Toutefois, dans notre observation, on a pu remarquer des façons différentes de trouver les savoirs.

Dès le départ, des échanges ont lieu entre les apprenant-es dans lesquels il est question des meilleures ressources d’apprentissage : cours en ligne, tutoriels vidéos, sites pratiques. Tout au long de la formation, les apprenant-es se partagent les documents pdf et les ressources documentaires qu’ils ou elles ont pu trouver. Mais le partage des ressources ne suffit pas à mettre les apprenant-es sur un pied d’égalité. Le nombre de contenus disponibles est bien sûr beaucoup plus développé en anglais. S’il serait peu prudent de se prononcer sur la qualité des contenus selon les langues, on peut au moins faire remarquer que les développeur-ses de langue français.es écrivent souvent en anglais les documentations, et que nombre d’outils informatiques n’ont de documentation – qui permet d’apprendre à les utiliser – qu’en anglais. Par conséquent, les personnes ne maîtrisant pas un minimum l’anglais peuvent être pénalisées – même si la plupart utilisaient les outils de traduction automatique intégrés dans leurs navigateurs Internet.

À ces prérequis s’ajoute la capacité à savoir ce qu’on cherche, elle aussi inégalement répartie. Il n’est en effet pas évident, que l’on soit confronté à un problème, qu’on n’arrive pas à appliquer une consigne, ou qu’on ait tout un projet à faire, de savoir concrètement ce dont on a besoin ou par où commencer. Ainsi, on observe des différences entre les personnes qui vont, si ce n’est réussir, au moins tenter de resituer un savoir dans un cours. Par exemple, il faut déjà un certain rapport à l’apprentissage – acquis ou non à l’école – pour penser une technologie (par exemple

MySQL) dans une perspective plus large (par exemple, un cours sur les bases de données), et par conséquent pour choisir de se lancer dans un cours en ligne englobant cette technologie.

« On va passer la formation à faire des tutos en fait. »

Lucas, 32 ans

Ces propos de Lucas témoignent du malaise parfois ressenti par les apprenant-es, qui ont pu exprimer une forme de dénigrement de la formation qui ne conçoit et n'offre aucun contenu d'apprentissage. Pasquier soulignait que ses enquêté-e-s appréciaient regarder des tutoriels sur leurs heures de loisirs : pour les nôtres, le visionnage de tutoriels semblait ne pas toujours être à la hauteur de leurs attentes, peut-être parce qu'il était justement associé à un temps de loisirs et à un contenu auquel tout le monde a accès.

Il n'y a donc plus de leçons d'un-e professeur-e au tableau... Mais chacun-e se retrouve à prendre une leçon d'un-e inconnu-e sur une vidéo. L'injonction à apprendre à apprendre consiste donc, pour cette étape, à devoir trouver un-e enseignant-e. Comme dit précédemment, la formation repose essentiellement sur le fait que d'autres personnes ou institutions ont créé du contenu d'apprentissage.

Comprendre

La deuxième étape de l'apprentissage consiste à comprendre les res-

sources trouvées. Ici aussi, la pédagogie active a un effet : par son absence de contrainte, elle agit sur le processus de compréhension

Extrait de journal de terrain (7 août) :

« Ça m'angoisse Sirine et Paul qui avancent à fond par copier-coller » me confie Charlotte. En fait, Sirine et Paul peuvent être beaucoup plus rapides que les autres pour finir certains projets ou exercices, car ils copient-collent des réponses, c'est-à-dire du code, sans nécessairement les comprendre, et parfois ça marche.

Que ce soit des projets que les apprenant-e-s ont à réaliser ou *a fortiori* des TP, il est possible de copier-coller du code trouvé sur Internet afin de réaliser l'exercice. L'ambiguïté réside dans le fait que s'il est demandé aux développeur-ses en herbe d'être « malin » – pour reprendre les termes d'un formateur – et efficace, il s'agit aussi de comprendre le code réutilisé. Or, il n'est pas nécessaire de comprendre le code pour que ce dernier fonctionne. Il est par exemple possible de faire de très nombreuses tentatives, sans nécessairement réfléchir à ce que chacune implique, jusqu'à ce que cela donne un bon résultat lorsque le code est exécuté. Ou il est aussi possible de copier-coller beaucoup de code et, lorsque cela ne fonctionne pas, de demander à quelqu'un d'un niveau plus avancé de le corriger. De sorte que certain-es, tout en ayant fini les exercices ou réalisé certains projets, n'ont pas forcément acquis toutes les connaissances que l'exercice était censé faire acquérir.

Les travaux de Bonnéry sur des dispositifs pédagogiques favorisant l'activation des élèves en fin d'école primaire et début de collège nous semblent particulièrement éclairer la formation observée. Pour faire le travail demandé par leur enseignant, les élèves n'ont pas besoin de comprendre les notions dont l'exercice est censé favoriser l'appropriation : ainsi, lorsque les élèves sont invités à colorier une carte en géographie pour s'approprier la notion de relief, il est tout à fait possible pour eux de colorier la carte sans avoir saisi ce à quoi cela renvoyait (Bonnéry, 2007). Dès lors, l'élève n'apprend pas, notamment parce que, selon l'auteur, il ne sait pas qu'il y a quelque chose à apprendre derrière cet exercice pratique : il n'a pas conscience que l'école propose un aller-retour entre concept et cas pratique, excepté s'il est « *complice des évidences scolaires* » (Bonnéry, 2009b, p. 16). Tout en étant basé sur la critique des dispositifs pédagogiques classiques, ce cadre d'apprentissage reproduit pourtant ses défauts : ne pas soulever les prérequis implicites nécessaires pour comprendre ou apprendre quelque chose et redevenir ainsi une école indifférente aux différences (Bourdieu, 1966). On peut ainsi reprocher à ces dispositifs pédagogiques d'être des « *fabriques passives* » d'inégalités (Bonnéry, 2009a, p. 15). C'est ce qui se joue ici dans les exercices proposés, qu'il est possible de faire sans réfléchir.

Dans le cadre observé cependant, les difficultés liées aux prérequis sont peut-être plus profondes, parce qu'il n'y a pas de questionnement des encadrant-es

sur les prérequis nécessaires pour faire un exercice. Par exemple, on nous demande de réaliser un « projet », durant plusieurs jours, dans lequel il faut analyser un jeu de données sur des logements Airbnb.

Extrait de journal de terrain (23 août) :

C'est étrange ce projet où on doit tirer des informations des données mais nous n'avons aucune notion de mathématiques enseignée. À peine un prof a-t-il fait un cours improvisé une matinée sur la médiane que seuls quelques-uns (Paul, Thuan) semblaient suivre. Ceux qui n'avaient pas les bases en mathématiques ne devaient pas réussir à le faire. Ce projet a été fait alors que nous n'avons quasiment aucune base dans les outils utilisés (Pandas²³ notamment) pour faire l'analyse de données. Avec l'ancienne prof, nous avons fait quelques exercices de Pandas, une fois, une journée courte, en juin.

De plus, dans la formation, il n'est pas question d'apprendre pourquoi utiliser certains outils plus que d'autres, quelle technique est la meilleure selon l'usage que l'on souhaite en faire, ou pour parler comme beaucoup de développeur-es, quelles sont les « bonnes pratiques ». Toutes ces dimensions de l'apprentissage ne sont pas abordées, bien qu'il arrive que ce type de question émerge chez les apprenant-es. Chaque outil appris l'est avec la justification qu'il est utilisé dans le monde du travail – ce à quoi on peut s'attendre, après tout, puisque le contenu

²³ Pandas est une librairie Python utilisée pour faire de l'analyse de données.

de formation est censé préparer à une certification professionnelle. Rien ne permet en tout cas de développer un regard critique sur ce qui est fait et la manière dont cela s'est fait durant la formation.

« Ici on fait des choses mais on n'apprend pas à nommer les choses, on ne sait pas comment ça s'appelle en fait ».

Audrey, 28 ans.

Cette citation d'enquêtée témoigne d'un autre aspect de l'apprentissage : la capacité à nommer ce qui a été appris, c'est-à-dire au fond à catégoriser des objets, des écrits, des outils, des techniques. Mais, comme l'enquêtée en témoigne, on n'apprend pas ou peu à nommer dans la formation observée : parfois, il est inscrit en intitulé d'un exercice le nom de ce qui est vu ensuite (par exemple, « récursivité »), mais cela n'est pas toujours le cas. Or, nommer, c'est aussi la voie d'accès à l'autonomie, puisque cela permet de chercher sur un moteur de recherche. La capacité à nommer est aussi au cœur de l'autorégulation dont il était question précédemment. On peut postuler ainsi que la pédagogie par projet favorise cet apprentissage sans nom, un apprentissage pratique, au détriment de savoirs théoriques. Tout se passe comme s'il ne fallait apprendre que de l'utilisation technique. Lucie Tanguy, ayant étudié les savoirs enseignés dans les lycées techniques il y a presque un demi-siècle, proposait alors une vision de l'enseignement de seconde classe : « *C'est dire que la technologie se présente non pas comme*

un savoir jugé socialement nécessaire à tous, mais comme un savoir réservé à une fraction de la jeunesse définie d'une manière négative, celle qui échoue dans l'apprentissage des disciplines scientifiques et littéraires » (Tanguy, 1983, p. 338). Les travaux de Tanguy ont montré comment l'école reproduisait une division des savoirs et distribuait inégalement les connaissances : ainsi, on apprend aux uns des savoirs « *scientifiques, universels, abstraits* » et aux autres des savoirs « *techniques, particuliers, concrets* » (*ibid.*), et on oppose et hiérarchise théorie et pratique. La formation semble ainsi préparer à une position professionnelle subalterne, dans laquelle on applique mais on ne discute pas.

Mémoriser

En comparaison avec des formations plus classiques, le programme de formation se révèle peu réaliste : il est prévu d'aborder énormément de technologies²⁴. Par conséquent, il est proposé aux apprenant·es de travailler deux jours sur une chose, puis deux jours sur une autre, la plupart du temps sans que les étapes de difficulté n'aient été pensées. Ceci peut être une cause possible du fait que certain·es se plaignent en cours de formation de ne pas se rappeler ce qu'ils avaient vu en début de formation, faute de pratique. La formation donne donc

²⁴ Ce que remarquent tou-te-s les professionnel-le-s de l'informatique qui ont regardé le programme de formation.

l'impression d'une initiation permanente. Aux plaintes, on répond que c'est à chacun de travailler sur son temps personnel afin de continuer à s'entraîner et maintenir les savoirs acquis. À nouveau, il s'agit de rendre l'apprenant-e autonome dans l'étape de mémorisation, et de fait, responsable s'il oublie. Du reste, la formation observée met très peu voire pas du tout l'accent sur le fait de retenir ce qui a été appris. Ceci concorde avec les recherches en sociologie de l'éducation qui montrent que les dispositifs pédagogiques récents mettent moins l'accent sur la mémorisation : au contraire, ils se concentrent sur la compréhension (Bonnéry, 2009a).

Expliquer aux autres

L'apprenant-e est aussi fortement incité-e à endosser le rôle du professeur et à expliquer et aider les autres. Malheureusement, jouer à l'enseignant se révèle difficile. Il est en effet délicat de trouver le juste milieu entre ne pas aider assez et donner la réponse : l'apprenant-e qui enseigne doit donner l'indice qui mette sur la bonne voie et qui guide, mais qui laisse encore la personne aidée réfléchir. De fait, la pédagogie active semble nier implicitement des savoirs pédagogiques acquis par le professeur.

Extrait de journal de terrain (21 juin) :

Parfois, on explique de cinq façons différentes quelque chose, et la personne ne comprend toujours pas. Parfois, on se rend compte que la personne n'a

pas du tout les bases permettant de comprendre l'exercice ; mais on met du temps à le découvrir.

Cet aspect est d'autant plus difficile que l'apprenant-e n'a de fait pas la position de l'enseignant, position qui légitime certaines actions, qui peuvent être pensées comme directives ou intrusives. Par exemple, lorsqu'il faut poser des questions à la personne aidée afin de comprendre ce qu'elle a compris ou *a fortiori* ce qu'elle n'a pas compris, ou lorsqu'il faut lui suggérer de revoir certains exercices de base, qui sont parfois des exercices de début de formation. Il faut oser et se permettre de dire à quelqu'un qu'il faudrait réviser les TP réalisés la première semaine de formation lorsqu'on en est au quatrième mois.

Extrait de journal de terrain (18 décembre) :

Omar est en train de faire le projet de Marion qui lui a demandé de l'aide, tandis qu'elle est en train de parler ailleurs. Elle veut changer les colonnes d'un dataframe pour avoir autre chose. Omar ne comprend pourquoi elle veut autre chose mais a compris ce qu'elle voulait alors il lui fait.

De plus, comme cet extrait de journal de terrain le souligne, il est difficile pour les apprenant-es de ne pas tomber dans l'autre écueil : faire le travail de la personne que l'on souhaite aider. Or, il arrive assez régulièrement qu'un homme prenne l'ordinateur d'une femme pour résoudre le problème auquel elle est confrontée.

Évaluer

L'autoapprentissage implique aussi de s'évaluer seul. Ainsi, une fiche d'autoévaluation est distribuée dans laquelle les apprenant-es sont invité-es à noter pour chaque compétence s'ils sont en mesure d'imiter, d'adapter ou d'améliorer. L'enseignant note ensuite si l'apprenant-e est capable d'imiter, d'adapter ou d'améliorer selon son point de vue. Toutefois, ceci est réalisé sans que les apprenant-es sachent quel sens et quelle valeur lui donner. En effet, l'apprenant-e ne sait pas toujours à quoi correspondent les compétences proposées, ni dans quelle mesure il les a acquises. L'enseignant-e n'a généralement pas vu l'apprenant-e réaliser quelque chose puisque la plupart des projets sont à faire en groupe et en autonomie. Il pose son jugement après avoir posé à l'apprenant-e la même question que celui-ci/celle-ci se pose lorsqu'il remplit lui-même l'évaluation, par exemple « est-ce que tu es capable d'importer des données ? ». Or, l'absence d'un retour extérieur ayant de la valeur peut influencer l'évaluation de soi de chacun des apprenant-es. Par exemple à résultats équivalents, dans des domaines considérés comme masculins, les hommes ont tendance à considérer leurs compétences plus élevées que les femmes ne le font ; mais cet écart dans l'autoévaluation se voit modéré lorsqu'il y a une évaluation extérieure (Correll, 2001).

« Il [Un apprenant] n'a pas la chance d'avoir fait de la programmation avant

comme moi, ni de comprendre vite comme Antoine et Bastien, ni d'être bosseurs comme Audrey et Lola. »

Omar, 28 ans.

Aussi, les perceptions genrées des apprenant-es sont les mêmes que celles observées chez les enseignants en école primaire : les femmes ont été perçues comme plus travailleuses, moins en proie au don, au génie, tandis que les hommes ont, pour reprendre les mots d'Omar, « *la chance* » de « *comprendre vite* » (Duru-Bellat, 2004). Ainsi, il serait intéressant d'objectiver les différences de perception des apprenant-es par les encadrant-es en fonction, notamment, du sexe des apprenant-es, dans le contexte de pédagogie relevant du code intégré : l'absence d'évaluation classique renforce-t-elle ou diminue-t-elle les processus de stéréotypage ?

Conclusion

« *Si on m'explique, je ne vais pas comprendre* » a répondu l'un²⁵ de nos enquêté-e-s afin de justifier l'intérêt d'un apprentissage en autonomie : cette phrase peut renvoyer à l'idée que l'on apprend toujours seul, parce que l'apprentissage est un processus interne. Les sciences de l'éducation nous rappellent qu'il faut ainsi distinguer l'acte d'apprendre de l'acte d'enseigner : on ne peut pas

²⁵ Propos d'un enquêté (A., 26 ans, en formation « développement data »).

apprendre quelque chose à quelqu'un, on ne peut que lui enseigner et en favoriser l'apprentissage (Carré, 2015 ; Vermersch, 2012). L'idée d'« apprendre par soi-même », corollaire de l'idée d'« apprendre à apprendre », n'a dès lors pas de sens en tant que phénomène nouveau, puisque la seule chose que l'on peut modifier, c'est l'acte d'enseigner pour favoriser l'apprentissage. De même, le fait que ce type de pédagogie soit présentée comme innovante est en soi à interroger puisque l'invention d'un dispositif pédagogique de ce type ne date pas d'aujourd'hui.

On a pu observer un grand décalage entre les discours vantant la méthode d'apprentissage de la formation observée, peignant des apprenant-es actif-ves, sur la route de l'autonomie, dans un environnement bienveillant, et la réalité d'un apprentissage laborieux et peu effectif. Il apparaît qu'« autonomie » est ici un joli mot pour dire le manque de moyens investis dans les ressources mises à disposition des apprenant-es. Elle permet de légitimer un dispositif pédagogique en rupture avec l'enseignement traditionnel mais dont les résultats en termes d'apprentissage apparaissent douteux. Puisque les pédagogies de l'autonomie permettent à chacune d'aller à son rythme, elles autorisent une variabilité très forte dans les rythmes d'apprentissage de chaque membre du groupe, favorisant un apprentissage à deux vitesses dans un même groupe. Toutefois, il est à noter que la plupart des apprenant-es ont obtenu leurs certifications professionnelles, bien que le contenu de formation prévu n'ait pas été réalisé puisque plu-

sieurs contenus n'ont pas été abordés et beaucoup ont été survolés.

Six mois après la formation, seules deux personnes avaient trouvé un emploi dans l'informatique. Avant même d'avoir l'occasion de mettre à l'épreuve leurs compétences en programmation, les enquêté-e-s se voient confronté-es à des discours pointant leur niveau de diplôme – trop faible pour trouver un emploi dans l'informatique –, leur manque d'expérience professionnelle, ou encore leur âge – trop élevé pour certain-es.

Après avoir mis l'accent sur la mise en pratique de la pédagogie « active », il nous semble qu'il serait pertinent de s'intéresser à ce qui semble aller de pair avec : l'esprit d'entreprise qui pénètre le domaine scolaire, y compris l'enseignement supérieur (Chambard, 2020 ; Clément, Dreux, Laval & Vergne, 2011). Dans ses travaux sur l'apprentissage de l'entrepreneuriat dans l'enseignement supérieur, Chambard souligne justement la velléité d'utiliser des formes pédagogiques dites « alternatives » dans ces formations – velléité car les formes scolaires classiques reprennent vite le dessus.

Enfin, cette formation à destination de chômeur-ses, par sa logique de projet, par son objectif d'insertion, par son individualisation de l'apprentissage, s'inscrit aussi tout à fait dans les dispositifs d'insertion décrits par Mauger (Mauger, 2001), et, à ce titre, mériterait d'être analysée comme un dispositif d'inculcation d'habitus adapté à l'entreprise néolibérale.

Bibliographie

Bacqué M.-H. & Biewener C. (2015). *L'empowerment, une pratique émancipatrice ?* Paris : La Découverte/Poche.

Benedetto-Meyer M., Boboc A. & Metzger J.-L. (2019). « Se former avec le numérique : entre exigence d'autonomie et quête de régulation ». *Communication & Organisation*, 56, pp. 47-61.

Bernstein B. (1975). *Langage et classes sociales : codes socio-linguistiques et contrôle social*. Paris : Éd. de Minuit.

Bernstein B. (1975). « Class and Pedagogies: Visible and Invisible ». *Educational Studies*, vol. 1, n° 1, pp. 23-41.

Boboc A. & Metzger J.-L. (2019). « La formation continue à l'épreuve de sa numérisation ». *Formation emploi*, 145, pp. 101-118.

Boboc A. & Metzger J.-L. (2019). « La formation continue numérisée face à ses discontinuités ». *Lien social et politiques*, n° 81, pp. 230-252.

Boboc A. & Metzger J.-L. (2016). « La formation professionnelle à distance à la lumière des organisations capacitantes » [en ligne]. *Distances et Médiations des Savoirs*, n° 14 [URL : <http://journals.openedition.org/dms/1447>].

Boltanski L. & Chiapello È. (1999). *Le Nouvel Esprit du capitalisme*. Paris : Gallimard.

Bonnéry S. (2009). « Scénarisation des dispositifs pédagogiques et inégalités d'apprentissage ». *Revue française de pédagogie*, n° 167, pp. 13-23.

Bonnéry S. (2007). *Comprendre l'échec scolaire : élèves en difficultés et dispositifs pédagogiques*. Paris : Dispute.

Bonnéry S. (2009). « Chapitre III. Contenus, pratiques pédagogiques et échec scolaire ». In M. Duru-Bellat & A. van Zanten. *Sociologie du système éducatif. Les inégalités scolaires*. Paris : Presses Universitaires de France, pp. 149-165.

Bourdieu P. (1966). « L'école conservatrice. Les inégalités devant l'école et devant la culture ». *Revue française de sociologie*, vol. 7, n° 3, pp. 325-347.

Carré P. (2015). « De l'apprentissage à la formation. Pour une nouvelle psychopédagogie des adultes ». *Revue française de pédagogie*, n° 190, pp. 29-40.

Chambard O. (2020). *Business model : l'université, nouveau laboratoire de l'idéologie entrepreneuriale*. Paris : La Découverte.

Clément P., Dreux G., Laval C. & Vergne F. (2011). *La nouvelle école capitaliste*. Paris : La Découverte.

Collet I. (2006). *L'informatique a-t-elle un sexe ? Hackers, mythes et réalités*. Paris : Harmattan.

Correll S. J. (2001). « Gender and the Career Choice Process: The Role of Biased Self Assessments ». *American Journal of Sociology*, vol. 106, n° 6, pp. 1691-1730.

Dupuy C. & Sarfati F. (2022). *Gouverner par l'emploi. Une histoire de l'École 42*. Paris : PUF.

Dürler H. (2015). *L'autonomie obligatoire : sociologie du gouvernement de soi à l'école*. Rennes : Presses universitaires de Rennes.

Duru-Bellat M. (2004). *L'École des filles : quelle formation pour quels rôles sociaux ?* Paris : L'Harmattan.

Duru-Bellat M., Farges G. & van Zanten

- A. (2018). « Chapitre 6. Les programmes, les pratiques pédagogiques et les normes d'excellence ». In M. Duru-Bellat, A. van Zanten (dir). *Sociologie de l'école*. Paris : Armand Colin. pp. 124-147.
- Govaerts S. & Grégoire J. (2006). « Chapitre VIII. Motivation et émotions dans l'apprentissage scolaire ». In B. Galand & E. Bourgeois. *(Se) motiver à apprendre*. Paris : Presses Universitaires de France, pp. 97-106.
- Granjon F., Lelong B. & Metzger J.-L. (2009). *Inégalités numériques : clivages sociaux et modes d'appropriation des TIC*. Paris : Hermès.
- Grignon C. (1971). *L'Ordre des choses, les fonctions sociales de l'enseignement technique*. Paris : Ed. de Minuit.
- Labarthe F. & Delorme V. (2019). « La Grande École du Numérique : en quête d'empowerment ? Une approche par les capacités » [en ligne]. *Terminal. Technologie de l'information, culture & société*, n° 125-126 [URL : <http://journals.openedition.org/terminal/5370>].
- Lahire B. (2001). « La construction de l'«autonomie» à l'école primaire : entre savoirs et pouvoirs ». *Revue française de pédagogie*, vol. 135, n° 1, pp. 151-161.
- Leroy G. (2017). « L'influence de l'éducation nouvelle sur les pratiques contemporaines de l'école maternelle française ». *Spécificités*, vol. 10, n° 1, pp. 61-86.
- Mauger G. (2001). « Les politiques d'insertion. Une contribution paradoxale à la désstabilisation du marché du travail ». *Actes de la Recherche en Sciences Sociales*. vol. 136-137, pp. 5-14.
- Mons N., Duru-Bellat M. & Savina Y. (2012). « Modèles éducatifs et attitudes des jeunes : une exploration comparative internationale ». *Revue française de sociologie*, vol. 53, n° 4, pp. 589-622.
- Mosconi N. (1999). « Les recherches sur la socialisation différentielle des sexes à l'école ». *Débats Jeunesses*, vol. 4, n° 1, pp. 85-116.
- Pasquier D. (2019). « Les pratiques numériques en milieu populaire ». *Études*, Juin, n° 6, pp. 51-60.
- Raab R. (2016). « Le paradoxe de l'autonomie en contexte scolaire » [en ligne]. *Éducation et socialisation*, n° 41 [URL : <http://journals.openedition.org/edso/1663>].
- Simon R. W. (2014). « Sociological Scholarship on Gender Differences in Emotion and Emotional Well-Being in the United States: A Snapshot of the Field ». *Emotion Review*, 6(3), pp. 196-201.
- Tanguy L. (1983). « Les savoirs enseignés aux ouvriers ». *Sociologie du travail*. Vol. 25, n° 3, pp. 336-354.
- Vandenbunder J. (2015). « Peut-on enseigner l'art ? Les écoles supérieures d'art, entre forme scolaire et liberté artistique ». *Revue française de pédagogie*, n° 192, pp. 121-134.
- Vermersch P. (2012). *Explicitation et phénoménologie : vers une psychophénoménologie*. Paris : Presses universitaires de France.
- Vicente M. (2019). « La Grande École du Numérique : les paradoxes d'une politique de promotion des formations techniques centrées sur l'apprentissage du code informatique ». *Lien social et politiques*, n° 81, pp. 212-229.
- Zune M. (2006). « De la pénurie à la mobilité : le marché du travail des informaticiens ». *Formation emploi*, n° 95, pp. 5-24.



*Imprimé dans les ateliers d'impression du CNAM
sur un papier agréé FSC/PEFC respectueux de l'environnement.*

Cahiers d'histoire du Cnam

L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques

coordonné par Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard

Dossier : L'informatique entre à l'école : vers une histoire de l'enseignement des sciences et techniques informatiques

Cédric Neumann, Camille Paloque-Bergès et Loïc Petitgirard – Introduction : « *Jalons pour une histoire de l'enseignement de l'informatique en France* »

Pierre Lescanne – Témoignage : « *Prenons la main de la mémoire. Ma découverte de l'informatique (1964-1973)* »

Gilles Blain, Gérard Nowak et Didier Vaudène – Enquête d'acteurs : « *Une pratique active de l'informatique à l'école* »

Jean-Noël Lafargue – Fonds d'archives : « *Archéologie d'un carton de l'Informatique pour tous* »

Antoine Meyer et Simon Modeste – « *Rôle d'un logiciel dans la transposition didactique du concept d'algorithme : le cas du logiciel AlgoBox en France et des programmes du lycée entre 2009 et 2019* »

Isabelle Astic – Fonds d'archives : « *Le Goupil G3, point de basculement de l'industrie de la micro-informatique en France* »

Loïc Petitgirard et Camille Paloque-Bergès – Article de synthèse : « *La micro-informatique, outil et enjeu d'innovations dans les formations du Cnam (1970-1990)* »

Loïc Petitgirard – Témoignage : « *Le Cnam, berceau de la micro-informatique. Entretien avec Bruno France-Lanord* »

Cédric Neumann – « *La naissance de l'informatique à l'École centrale à la fin des années 1960 : un enseignement au service de la formation de managers* »

Michaël Vicente – « *L'utilisation des tests de sélection des informaticiens des années 1960 aux années 1990 : une hiérarchisation des formations et des publics* »

Mathilde Krill – « *“Si quelqu'un m'explique, je vais pas comprendre.” Une formation d'informatique face à la pédagogie de l'autonomie* »

● vol.15

2022 / Premier semestre
(nouvelle série)

ISSN 1240-2745