

numéro

7

1024

B U L L E T I N

de la société informatique
de France

novembre

2015

COMITÉ DE RÉDACTION

SYLVIE ALAYRANGUES
Université de Poitiers

OLIVIER BAUDON
Université de Bordeaux

COLIN DE LA HIGUERA
Université de Nantes

JEAN-PAUL DELAHAYE
Université Lille 1

OLIVIA DUMAS
Académie de Caen

CHRISTINE FROIDEVAUX
Université Paris-Sud

THIERRY GARCIA
*Université de Versailles Saint-Quentin-
en-Yvelines*

MARIE-CLAUDE GAUDEL
Université Paris-Sud

CLAUDIA MARINICA
Université de Cergy-Pontoise

PHILIPPE MARQUET
Université Lille 1

EUNIKA MERCIER-LAURENT
Innovation 3D et Université Lyon 3

DENIS PALLEZ
Université Nice Sophia Antipolis

JEAN-MARC PETIT
INSA Lyon

VINCENT RIBAUD
Université de Bretagne Occidentale

ÉRIC SOPENA, *Université de Bordeaux, rédacteur en chef*

Contact : 1024@societe-informatique-de-france.fr



*Cette œuvre est mise à disposition sous licence Attribution - Pas de Modification 4.0 France.
Pour voir une copie de cette licence, visitez*

<http://creativecommons.org/licenses/by-nd/4.0/deed.fr>

*ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California,
94041, USA.*

SOCIÉTÉ INFORMATIQUE DE FRANCE

Institut Henri Poincaré, 11 rue Pierre et Marie Curie, 75231 Paris Cedex 05

Prix public : 32 € (adhérents SIF : -30%)

Directeur de la publication : Jean-Marc Petit

ISSN : 2270-1419

Couverture : d'après une maquette réalisée par Lollygraph.com.

SOMMAIRE DU N° 7



SIF

Mot du président, <i>Jean-Marc Petit</i>	3
Actualité de la SIF, <i>Sylvie Alayrangues</i>	7
Class'Code : des idées qui vont devenir une réalité, <i>Colin de la Higuera</i>	15

SCIENCE

Comment simuler efficacement les interactions lumière/matière en synthèse d'image, <i>Adrien Gruson et Kadi Bouatouch</i>	29
---	----

HISTOIRE

Le système LSE, <i>Jacques Baudé</i>	41
--	----

ENSEIGNEMENT

Lettre ouverte à Mme Najat Vallaud-Belkacem, <i>Maurice Nivat</i>	57
Enseigner l'informatique à des linguistes (filières TAL), <i>Isabelle Tellier</i>	61
Rapport d'activité de la formation ISN dans l'académie de Montpellier de septembre 2011 à juin 2015, <i>Olivier Cogis, Jérôme Palaysi et Richard Terrat</i>	65

REGARDS

Organisation du travail à l'âge informatique, <i>Laurent Bloch</i>	81
--	----

INFORMATIQUE ET SOCIÉTÉ

Qu'est-ce que l'intermédiation algorithmique ? <i>Stéphane Grumbach</i>	93
---	----

ANALYSES

L'architecture x86 est-elle sûre ? <i>Laurent Bloch</i>	113
---	-----

PRIX ET DISTINCTIONS

Des consensus tout sauf mous, <i>Anne-Marie Kermarrec</i>	119
Michel Raynal élu membre de <i>Academia Europaea</i>	123

RÉCRÉATION

Cinquante boîtes fermées, <i>Jean-Paul Delahaye</i>	125
---	-----



Mot du président

Jean-Marc Petit ¹

Chère adhérente, cher adhérent,

Les événements tragiques que nous traversons frappent chacun de nous et renforcent l'importance de l'éducation de nos enfants pour éclairer leur conscience, les inciter à raisonner par eux-mêmes afin qu'ils puissent durant toute leur vie porter un regard critique sur le monde, loin des positions dogmatiques des uns ou des autres.

La SIF partage l'ambition du ministère de l'Éducation nationale (MEN) de former des citoyens éclairés, garants de l'unité de la République française et de ses valeurs. Nous partageons leur volonté de voir nos concitoyens formés convenablement aux humanités et aux sciences, c'est-à-dire principalement la langue française, l'instruction civique, la philosophie, les langues étrangères, l'histoire-géographie pour les humanités, et les mathématiques, la physique-chimie, la biologie et l'informatique pour les sciences.

Quand la SIF se réjouit de l'introduction de l'informatique dans les programmes du primaire et du collège ², certains pourront y voir un combat corporatiste, avec de sombres questions de postes et de positions à prendre. La SIF y voit par-dessus tout une formidable opportunité de formation de nos enfants au monde dans lequel ils vivent – pas le monde dans lequel vivaient leurs parents ou leurs grands-parents – mais le leur, celui où l'informatique a pénétré toutes les strates de leur vie quotidienne.

1. Président de la Société informatique de France, professeur à l'INSA de Lyon, LIRIS (UMR 5205 CNRS), jean-marc.petit@insa-lyon.fr, [twitter : @calixtepetit](https://twitter.com/calixtepetit).

2. Voir par exemple <http://www.societe-informatique-de-france.fr/2015/11/sif-men-enseignement-informatique/>

Inculquer les valeurs de la République ne peut pas se faire au XXI^e siècle comme au XX^e, et nul doute que l'enseignement de l'informatique comme discipline scientifique et technique a un rôle crucial à jouer. Pourtant, la discipline informatique en tant que telle n'est toujours pas présente dans le secondaire, bloquée on ne sait où par on ne sait qui. Oui, elle perturbe l'ordre établi des disciplines du XX^e siècle, car elle apporte beaucoup en donnant des clés pour mieux appréhender le monde, contribuant à développer le libre arbitre de nos concitoyens.

Devenir professeur d'informatique dans le secondaire ne demande ni plus ni moins d'effort et de labeur que devenir professeur de mathématiques ou de français. En la matière, il ne peut pas y avoir de magie, de miracle, et la position du MEN qui consiste à confier cet enseignement stratégique aux professeurs de mathématiques et de technologie est une erreur qui ne pourra pas passer à l'échelle du pays.

À l'heure de la promotion au plus haut niveau de l'État du « grand plan pour le numérique », nous avons demandé au MEN – dans un esprit de cohérence – de créer cette discipline Informatique dans le secondaire, ce qui ouvrirait la voie à la création d'un corps de professeurs d'informatique formés comme ceux des autres disciplines.

Gageons que cette position trouve enfin un écho favorable au plus haut niveau de l'État et qu'elle soit réglée politiquement, afin de mettre un terme aux atermoiements des différentes forces en présence au sein du MEN.

o O o

L'actualité de la SIF est par ailleurs très riche, comme vous pouvez le constater avec la « *newsletter* » et la rubrique « Actualité » du bulletin 1024 qui vous donnent les éléments factuels sur notre engagement au quotidien. Je souhaite maintenant mentionner deux actions majeures que nous avons démarrées récemment.

Depuis la transformation de SPECIF en SIF en 2012, nous avons pour objectif légitime de devenir ARUP (Association reconnue d'utilité publique). Au-delà des aspects fiscaux, il s'agit d'un label important qui donnera une meilleure reconnaissance de nos actions. La constitution du dossier a été un long parcours, mené en interne par Elisabeth Muriasco, que nous remercions ici chaleureusement. La demande a été déposée au ministère de l'Intérieur le 19 octobre 2015 : le processus est maintenant lancé. Nous vous tiendrons informés des évolutions de ce dossier.

Sur le plan international, la SIF est devenue à l'automne 2015 le représentant officiel de la France à l'IFIP (*International Federation for Information Processing*). Un consortium – comportant trois instituts, Inria, CNRS INS2I, IMT, et deux associations, SEE et SIF – s'est constitué pour régler des problèmes d'ordre financier et relever les défis de la présence française dans les comités techniques de l'IFIP. Tout est à faire pour dynamiser notre participation à l'IFIP, rendre audibles les actions qui y sont menées et susciter de l'engouement auprès des plus jeunes. Nous nous

réjouissons de ce nouveau rôle confié à la SIF pour représenter la France au niveau international.

o O o

Ce numéro de 1024 devrait ravir celles et ceux qui aiment les bons mots et les points de vue sur l'informatique, à destination des académiques et des industriels. Concernant 1024, je tiens à saluer le remarquable travail du comité éditorial et de son rédacteur en chef, Éric Sopena. Naturellement, les articles publiés dans 1024 ne reflètent pas toujours la position officielle de la SIF, mais contribuent à l'expression de la diversité des points de vue et d'opinion. J'espère que vous apprécierez ce numéro à sa juste valeur, et que vous serez nombreux à nous rejoindre sur des actions concrètes pour agir et promouvoir notre discipline.

Je ne peux finir cet édito sans vous inviter une nouvelle fois à participer au congrès de la SIF (Strasbourg, 26-28 janvier 2016), qui débute avec une réunion du Conseil des associations et l'assemblée générale de l'association, dans laquelle nous aurons à élire de nouveaux membres du Conseil d'administration. C'est une occasion unique de rencontrer les acteurs d'aujourd'hui, de discuter d'Europe, de vie privée, du projet Class'Code et de bien d'autres choses encore !

Bonne lecture de ce septième numéro, en espérant vous retrouver nombreux à Strasbourg fin janvier.



Actualité de la SIF

Informations collectées par Sylvie Alayrangues

Une rubrique pour vous tenir informé de la vie de la SIF et vous inviter à y participer. Vous trouverez notamment ici des annonces de manifestations organisées ou soutenues par la SIF, un point sur les dossiers en cours, des appels à participation...

La SIF vous parle d'elle

Assemblée générale 2016

L'Assemblée générale de la SIF aura lieu le mardi 26 janvier 2016, à Strasbourg, au début du congrès annuel de la SIF. Comme chaque année, ce sera l'occasion de revenir sur l'année écoulée et de parler des projets pour l'année à venir. Les adhérents seront également appelés à voter pour renouveler une partie du Conseil d'administration. Surveillez vos boîtes mail, les professions de foi des candidats y arriveront très bientôt.

Si vous ne pouvez pas faire le déplacement, vous pouvez naturellement donner procuration.

Du côté du Conseil scientifique

Le compte-rendu du dernier conseil scientifique est accessible en ligne sur le site de la SIF¹.

Le premier séminaire des doctorants en informatique, initié par le Conseil scientifique, aura lieu le 29 avril 2016 à l'institut Henri Poincaré.

Toutes les informations utiles seront bientôt accessibles sur :

www.societe-informatique-de-france.fr/sdoc/

1. www.societe-informatique-de-france.fr/wp-content/uploads/2013/10/CR_CS-50ct2015.pdf

La SIF et les associations en informatique

Une nouveauté cette année, la SIF offre à chaque association faisant partie du conseil des associations deux bourses pour permettre à des doctorants de leur choix de participer au congrès. Chaque bourse comprend : inscription, hébergement (en chambre *twin*) et repas de midi.

D'autre part, le congrès 2016 de la SIF débutera, le mardi 26 janvier à 14h, par une rencontre du Conseil des associations.

Demande de reconnaissance d'utilité publique

La SIF a vocation à devenir une association reconnue d'utilité publique (ARUP). Construire le dossier a exigé un travail de titan². Il a pu être déposé courant octobre.

La SIF vous propose

Congrès 2016 : 26-27-28 janvier

Le congrès 2016 se déroulera à Strasbourg fin janvier 2016. Le thème retenu cette année est « Informatique et Europe »

Le 27 janvier, il sera question d' « informatique et vie privée en Europe ». Après l'accueil par des représentants de l'université de Strasbourg et de son laboratoire d'informatique, nous rentrerons dans le vif du sujet avec une conférence invitée de François Pellegrini (universitaire) sur la « liberté à l'ère du numérique ». La matinée se poursuivra par une intervention de Gwendal Legrand (CNIL) sur l'« évolution de la protection des données personnelles en France et en Europe ». L'après-midi sera organisée autour d'une table ronde sur le thème « sécurité et vie privée », avec Tris Acatrinei (industrielle, journaliste), Sébastien Gambs (universitaire), et Yannig Raffanel (OpenClassrooms).

Le 28 janvier, les débats tourneront essentiellement autour de la « recherche en informatique en Europe ». Véronique Cortier (académique) témoignera sur les « *ERC grant* » et parlera également de « vote électronique ». Après le déjeuner, aura lieu une rencontre avec les associations européennes : ACM Europe, Informatics Europe et IFIP.

Ce congrès comportera également des temps forts pour la vie de la SIF : rencontre des associations et assemblée générale, lors de la première journée, annonce officielle et remise du prix de thèse Gilles Kahn le 28 janvier au matin avec les exposés des lauréats et, pour conclure le congrès, une conférence invitée de Sophie de Quatrebarbes sur « Class'Code, une initiative de la SIF ».

2. Vous trouverez ici une idée du parcours à suivre pour qu'une association puisse être reconnue ARUP : www.interieur.gouv.fr/A-votre-service/Mes-demarches/Associations/Formalites-administratives-d-une-association/Partenariat-d-une-association-avec-les-pouvoirs-publics/Association-reconnue-d-utilite-publique.

Journée Sciences & Media 2016

La SIF a rejoint cette année la Société chimique de France, la Société française de physique, la Société française de statistique, la Société de mathématiques appliquées et industrielles et la Société mathématique de France, pour organiser la troisième édition de la journée « Sciences & Media ». Cette journée se veut un point de rencontre et d'échanges entre scientifiques et passeurs de sciences.

L'édition 2016 se déroulera le 1^{er} février à la préfecture de Paris. Il sera question des outils médiatiques pour parler de sciences aux enfants et aux adolescents :

« La formation à la science, y compris à la démarche scientifique, constitue un élément important de la construction intellectuelle des jeunes citoyens, en particulier à notre époque où sciences et technologies évoluent rapidement et prennent une place grandissante dans le débat public. L'information via les médias vient en complément de l'école, elle permet d'aborder les sciences avec un rythme différent, d'en donner le goût et de favoriser des vocations. Elle est également utile aux enseignants et formateurs.

Au cours de cette journée, nous essaierons de répondre à des questions telles que : Comment communiquer selon les âges ? Comment prendre en compte la transition enfance/adolescence ? Quels médias, quels contenus sont les plus adaptés ? Comment les jeunes s'approprient-ils les médias ? Comment lutter contre les stéréotypes (genrés, sociaux...) qui constituent des freins à l'intérêt de tous, et en particulier des jeunes filles, pour les sciences ?³ »

Des intervenants de différents horizons : sociologues, scientifiques, représentants de la presse écrite, du monde de l'édition, du monde associatif, collectif de scientifiques et d'artistes, auteurs de science-fiction, viendront confronter leurs expériences et leurs approches. Les échanges se poursuivront également dans le hall de la préfecture, qui accueillera une exposition animée par différents acteurs de ces formes de médiation à destination des jeunes.

Le programme détaillé est accessible sur le site web de l'événement :

www.sciencesetmedia.org

« Save the date » : Journée recherche et innovation, 23 mars 2016

En 2016, la SIF vous propose une journée recherche sur le thème : « les défis de l'Internet des objets ». De plus amples renseignements seront bientôt accessibles via le site internet de la SIF.

3. <http://www.sciencesetmedia.org/>

La SIF soutient

Le concours Castor

Comme chaque année, le concours Castor s'est déroulé en novembre. Suite aux tragiques événements du 13 novembre, la plateforme est restée ouverte un peu plus longtemps que prévu initialement et à l'heure de parution de ce bulletin, les résultats et statistiques de participation ne sont pas encore connus.

D'ici peu, toutes ces informations ainsi que les annales du concours 2015 seront accessibles sur le site⁴.

À noter, une nouveauté pour cette année 2015 : pour la première fois, le concours était ouvert aux élèves de CM1 et CM2.

Le prix Novelli

Comme chaque année depuis sa création, la SIF était présente dans le jury du prix Bernard Novelli, qui fait partie des trophées Tangente et a pour but de récompenser des lycéens auteurs de projets informatiques autour du jeu. Le but était ici de concevoir un jeu ou une IA de jeu mathématique et/ou logique.

Le suspens sera bientôt levé puisque le prix sera décerné le mercredi 9 décembre à 16h.

La SIF prend position et agit

En enseignement

Vous en avez déjà entendu parler et ce n'est que le début (voir l'article de Colin de la Higuera dans ce numéro). Le projet Class'Code⁵, financé par le PIA (Programme d'investissements d'avenir), a été officiellement lancé en septembre 2015 devant une cinquantaine de personnes, regroupant notamment l'ensemble des partenaires actuels du projet.

Pour mémoire, il s'agit d'un projet ambitieux rassemblant des partenaires de divers horizons et visant à former les 300 000 personnes dont la France a besoin pour initier les enfants à l'informatique et au code dans le temps périscolaire. La spécificité du projet : proposer un MOOC hybride, mêlant apprentissage à distance et regroupements en région pour accompagner l'apprentissage des futurs 300 000 formateurs. La formation proprement dite commencera en septembre 2016.

Notre rôle à nous, informaticiens de la SIF : suivre le MOOC pour pouvoir apporter notre expertise scientifique aux autres apprenants et les aider, notamment en participant aux regroupements, à « maîtriser la pensée informatique pour la transmettre ». Nous avons besoin de tous les talents.

4. www.castor-informatique.fr/

5. www.classcode.fr

Les prochaines journées pédagogiques de la SIF, en juin 2016, vous éclaireront sur ce qui peut être fait au sein de Class'Code et au-delà, pour aider à former ceux qui forment nos enfants. Si vous vous demandez comment participer à cette aventure, n'hésitez pas à nous contacter.

Contact pour la SIF : Colin de la Higuera, cdlh@univ-nantes.fr

La SIF continue bien évidemment de se battre sur le front de l'enseignement de l'informatique afin que cette science puisse être enseignée dans de bonnes conditions par des enseignants formés à cela. Pour cela, elle peut de plus en plus s'appuyer sur des rapports établis par des autorités variées. Ainsi, récemment, l'OCDE a publié un rapport présentant les résultats de Pisa 2012 où le rôle et la place du « numérique » occupent une place importante. Comme vous pourrez le lire sur notre site⁶, les conclusions de ce rapport vont dans le sens de ce qu'a toujours défendu la SIF : une formation adéquate est nécessaire pour les enseignants et on ne peut se contenter d'enseigner les usages de l'informatique si l'on veut que les jeunes puissent réellement s'appuyer sur leurs connaissances dans leur vie professionnelle future.

La SIF et l'EPI ont également réagi aux diverses annonces à propos de l'enseignement de l'informatique et du numérique qui ont émaillé la rentrée scolaire pour que l'enseignement de l'informatique s'inscrive dans la durée⁷.

Suite à l'interview de Najat Vallaud-Belkacem dans 20 minutes⁸, la SIF a choisi de réagir aux propos de Madame la ministre sur Twitter (@SocInfoFr) et a lancé le hashtag #CapesInfoOui, largement repris.

Voici quelques réactions assez représentatives de nos « followers » :

Jean-Yves

« @najatvb The whole world knows that IT should be taught in schools except French. It's not very #FrenchTech ! #CapesInfoOui »

Luc

« La lecture et l'écriture aussi sont des activités transversales : pourquoi employer des personnes formées pour les enseigner ? #CapesInfoOui »

Sébastien

« Je croyais qu'on parlai[t] d'un CAPES d'info, pas de bureautique. #AuTempsPourMoi #CapesInfoOui @SocInfoFr @najatvb »

6. www.societe-informatique-de-france.fr/2015/09/connectes-pour-apprendre-les-eleves-et-les-nouvelles-technologies/

7. <http://www.societe-informatique-de-france.fr/2015/10/enseigner-linformatique-il-est-temps-de-sinscrire-dans-la-duree/>

8. <http://www.20minutes.fr/societe/1710175-20151015-numerique-ecole-2018-100-colleges-connectes>

La SIF ne se contente évidemment pas de réagir *a posteriori*. Elle continue à discuter en amont avec les différentes instances concernées. Notamment, elle a été reçue avec l'EPI en octobre par la « mission Monteil », nommée en mars dernier pour réfléchir à la politique numérique pour l'Éducation nationale. Vous pouvez lire le compte-rendu de cet entretien sur le site de l'EPI⁹. Suite au texte final remis par le Conseil supérieur des programmes (CSP) concernant les cycles 2, 3 et 4, la SIF était également reçue au ministère de l'Éducation nationale, début novembre, pour discuter de sa mise en œuvre et identifier les points de blocage. Vous pouvez retrouver les éléments clés de cet entretien sur le site de la SIF¹⁰.

La SIF participe également à divers événements, que ce soit sous forme d'interventions comme au colloque ARDM-CFEM¹¹ autour du thème « Mathématiques et informatique, quelles synergies dans l'enseignement ? »¹², ou via une participation à une table ronde comme lors du salon POSS (*Paris Open Source Summit*¹³).

La SIF poursuit bien évidemment les réflexions dans le cadre de divers groupes. Ainsi, depuis quelques mois, un groupe de travail a été mis en place à l'initiative de la SIF et de la SMF. Il réunit mensuellement des représentants de la SIF et des sociétés savantes de mathématiques et s'intéresse à l'enseignement de mathématiques liées à l'informatique au lycée. Par ailleurs, on ne présente plus le groupe ITIC-EPI-SIF dont la 26^e réunion est prévue le 12 janvier prochain¹⁴.

En recherche

Sur la plan international, la SIF a été sollicitée pour être le nouveau représentant officiel de la France à l'IFIP¹⁵ (*International Federation for Image Processing*). Pour mémoire, l'IFIP est une société savante internationale dans le domaine de l'informatique, au sein de laquelle chaque pays est représenté par une société savante nationale. La France est partie prenante de l'IFIP depuis ses débuts et de nombreux Français participent à ses activités. Mais depuis quelques années, la France n'avait plus de représentant officiel au sein de l'IFIP. Notre candidature, soutenue par un consortium composé, outre la SIF, d'Inria¹⁶, de l'INS2I¹⁷, de l'Institut Mines-Telecom¹⁸

9. www.enseignerlinformatique.org/2015/10/19/audience-de-lepi-et-de-la-sif-avec-le-recteur-jean-marc-monteil-et-alain-sere/

10. www.societe-informatique-de-france.fr/2015/11/sif-men-enseignement-informatique/

11. ARDM : Association pour la recherche didactique en mathématiques ; CFEM : Commission française pour l'enseignement des mathématiques.

12. <http://www.societe-informatique-de-france.fr/2015/10/colloquim-ardm-cfem/>

13. www.societe-informatique-de-france.fr/2015/11/poss-ens-info-et-libre/

14. Le compte-rendu de la 25^e réunion est accessible sur : www.enseignerlinformatique.org/2015/09/17/compte-rendu-de-la-25e-reunion-du-groupe-itic-epi-sif-2/

15. <http://www.ifip.org/>

16. www.inria.fr

17. www.cnrs.fr/ins2i/

18. www.mines-telecom.fr/

et de la SEE¹⁹, a été acceptée en octobre dernier. La France retrouve donc sa place et sa voix à l'assemblée générale de l'IFIP.

En médiation scientifique

La SIF a été appelée il y a plusieurs mois à rejoindre la SMF, la SMAI, la SFdS, la SIF, Inria et le CNRS (avec l'INSMI et l'INS2I) sur un projet de fondation « Mathématiques et Informatique » centrée sur les actions de médiation scientifique. Le but d'une telle fondation serait d'aider à financer des actions de médiation en mathématiques et en informatique sur l'ensemble du territoire. Les actions concernées par cette fondation pourraient être des actions à cheval entre les mathématiques et l'informatique mais aussi bien sûr des actions spécifiques à l'une des deux sciences. Une belle opportunité de partenariat qui ouvre des perspectives à tous les acteurs de la médiation de ces deux sciences.

Le blog Binaire²⁰ continue à nous faire voyager dans les vastes contrées de la science informatique, à raison d'un nouveau billet tous les 3 ou 4 jours, sur des thèmes aussi variés, que « Langue et langages », les licences utilisateurs, la programmation par contraintes, le concours Castor... Cette diversité, la qualité scientifique des articles, leur ton souvent humoristique, leur accessibilité à un public non spécialiste, ont conduit Binaire parmi les finalistes des *Golden Blog Awards* 2015 pour la catégorie « Sciences et recherche ». Il a été, *in fine*, coiffé au poteau par le blog « Passeur de sciences »²¹, également hébergé par le Monde, et on ne peut que féliciter ce blog scientifique généraliste de qualité qui a non seulement remporté cette catégorie mais a également été élu « meilleur Blog » pour l'année 2015, toutes catégories confondues.

Sur les enjeux de société

La SIF a relayé le communiqué de l'association ATALA (Association pour le traitement automatique des langues), association partenaire de la SIF, contre le développement des armes autonomes intelligentes à propos de la pétition « *Autonomous Weapons : an Open Letter from AI & Robotics Researchers* »²².

La SIF s'est aussi exprimée lors de la consultation sur le projet de loi sur le numérique en alertant sur certains articles. Elle a notamment appelé à voter contre l'article 9 concernant le libre accès aux publications scientifiques de la recherche publique et à voter en faveur du texte alternatif proposé par le CNRS²³. Quant à l'article 17,

19. www.see.asso.fr

20. <http://binaire.blog.lemonde.fr/>

21. www.passeurdsciences.blog.lemonde.fr/

22. www.societe-informatique-de-france.fr/2015/08/3187/

23. www.cnrs.fr/dist/z-outils/documents/Participation%20%C3%A0%20la%20consultation%20loi%20num%C3%A9rique.pdf

concernant les questions éthiques, la SIF a appelé à voter en faveur de la proposition de Max Dauchet, visant la création d'un Comité national d'éthique du numérique²⁴.

Dans un contexte difficile pour le marché de l'emploi, l'informatique et les mathématiques offrent des débouchés professionnels intéressants et parfois méconnus. Vous les aviez découverts dans la brochure sur les métiers des mathématiques et de l'informatique réalisée avec l'ONISEP pour la collection « Zoom métiers », vous pouvez maintenant faire plus ample connaissance via des vidéos. Les deux premières sont disponibles en ligne²⁵, d'autres les rejoindront bientôt. Ce travail est toujours mené conjointement par l'association Femmes et Mathématiques, la Société française de statistique (SFdS), la Société mathématique de France (SMF), la Société de mathématiques appliquées et industrielles (SMAI) et la Société informatique de France (SIF).

24. <https://www.republique-numerique.fr/consultations/projet-de-loi-numerique/consultation/consultation/opinions/section-1-protection-des-donnees-a-caractere-personnel/creer-un-comite-national-d-ethique-du-numerique-1>

25. www.societe-informatique-de-france.fr/2015/10/zoom-metiers-informatique-et-mathematiques/



Class'Code : des idées qui vont devenir une réalité

Colin de la Higuera^{1, 2}

Prologue

Informatique, numérique, code... Au moment d'écrire ces lignes, l'informaticien voit une réelle prise de conscience de l'importance pour l'école d'entrer dans le XXI^e siècle mais sent bien que le compte n'y est pas : ambition certes affichée, mais comment y arriver ? Comment se doter des enseignants formés pour l'enseignement de l'informatique (puisque c'est bien de cela qu'il s'agit) ?

Si la Société informatique de France, avec d'autres, n'hésite pas à participer au débat public, à rencontrer des partenaires institutionnels pour convaincre des bons choix à effectuer, elle doit également jouer un rôle d'acteur dans la transformation de la société qui est aujourd'hui nécessaire.

C'est le but que la SIF s'est fixé en participant à la mise en place du projet Class'Code³. C'est le but qu'elle doit maintenant se fixer en soutenant le déploiement de ce projet.

1. Professeur à l'Université de Nantes, Président des comités d'organisation et de pilotage de Class'Code.

2. J'en profite pour remercier toutes les personnes qui s'investissent sans compter dans le projet Class'Code, et en particulier Sophie de Quatrebarbes, chef de projet, et Thierry Viéville dont l'investissement est inversement proportionnel à la majesté des titres qu'il porte.

3. Son site : <https://classcode.fr/>; une présentation *grand public* : <http://binaire.blog.lemonde.fr/2015/10/05/classcode-cest-parti/>

L'identification de l'enjeu

La question n'est hélas pas nouvelle : depuis bien longtemps, lorsque l'on suggère que l'informatique pourrait être enseignée un peu plus sérieusement, une réponse (souvent implicite) est « mais nous n'avons pas les enseignants pour le faire ». Et quand on indique qu'il serait bon d'inclure un enseignement d'informatique dans les cours dispensés dans la formation des enseignants, ou qu'un CAPES (ou une agrégation) en informatique pourrait être créé pour justement avoir le corps d'enseignants capables de faire face à ces défis du XXI^e siècle, la réponse fuse immédiatement : « mais ce serait une erreur : il n'y a pas assez d'enseignement à pourvoir... la personne recrutée pour enseigner l'informatique serait très malheureuse, obligée de voyager continuellement de lycée en lycée pour gérer un service compliqué. »

Ce cercle vicieux semble éternel. Mais il peut se briser en introduisant dans le système, progressivement, les éléments pour que ça change. D'une certaine façon, le ministère de l'Éducation nationale est en train d'y contribuer en introduisant des enseignements d'initiation à l'informatique, à plusieurs niveaux, au collège et au lycée. Mais pour accélérer la fin de ce cercle vicieux il est souhaitable également d'ajouter des compétences informatiques aux professionnels de l'éducation qui pourront alors rassurer les pouvoirs publics sur l'existence d'éducateurs susceptibles de former les enfants et les adolescent-e-s.

Le besoin qui est apparu récemment de façon quasi consensuelle en France est celui de la formation au *code*. Nous analyserons dans ce qui suit les raisons de l'apparition de ce besoin, et ce en quoi l'informatique est une réponse adéquate au besoin exprimé. Puis comment on peut espérer former les dizaines de milliers d'éducateurs, de formateurs, d'animateurs du périscolaire ou de l'éducation populaire, d'enseignants qui pourront alors transmettre des connaissances en informatique aux enfants et aux jeunes adolescent-e-s.

Mais au fond, que veut-on enseigner ?

Numérique, code, informatique, programmation... les mots changent... mais les idées aussi.

L'informatique et le numérique

Depuis plusieurs années déjà, l'Éducation nationale nous explique qu'il convient de former (les enfants) *au numérique et par le numérique*. L'usage du substantif « numérique » interroge, d'emblée : dans les autres langues, c'est plutôt un adjectif qualificatif. Son statut de substantif impliquerait l'existence d'une réalité à elle seule, pas la qualification supplémentaire d'autre chose (le monde numérique, les arts numériques, les humanités numériques, l'économie numérique).

Milad Doueihi analyse de façon très pertinente les liens entre numérique et informatique⁴ : « *Si l'informatique a commencé comme branche des mathématiques, elle a rapidement trouvé son autonomie et son statut de science à part entière. Puis, chose relativement rare dans l'histoire des sciences, elle s'est transformée en industrie [...]. Fait unique, elle est également devenue, depuis au moins une vingtaine d'années, une culture. Et c'est bien cette spécificité culturelle, cette orientation sociale qui caractérise et en fin de compte définit en quelque sorte le numérique.* »

L'informatique, selon Milad Doueihi, est plutôt science et industrie. La Société informatique de France est dans cette ligne quand elle propose la définition suivante : « *L'informatique est la science et la technique de la représentation de l'information d'origine artificielle ou naturelle, ainsi que des processus algorithmiques de collecte, stockage, analyse, transformation, communication et exploitation de cette information, exprimés dans des langages formels ou des langues naturelles et effectués par des machines ou des êtres humains, seuls ou collectivement.*⁵ »

Le code, comme vecteur

Si le mot d'ordre était il y a encore deux ans celui de la formation au numérique, un nouvel acteur s'est introduit dans le débat : le code.

Le « code » est devenu un acteur majeur des débats, en France, en février 2013, avec l'irruption d'une vidéo de Barak Obama⁶. En un peu plus d'une minute, le Président des États-Unis mettait à mal toute la construction intellectuelle selon laquelle le citoyen numérique pouvait se contenter d'une formation aux usages. Cette position, jusque-là dominante en France, a explosé en quelques minutes.

Que disait exactement Barak Obama ?

- il rappelait que le leadership mondial de son pays devait aujourd'hui beaucoup à son industrie informatique qui transformait complètement la société ;
- il exhortait à apprendre à créer avec l'informatique : *don't play, build your app* ;
- il liait la capacité de créer avec les sciences, le besoin d'étudier la *computer science* en lui donnant une connotation forte en mathématiques et en sciences.

Obama emploie de façon variable les mots *code* et *computer science* : s'ils ne sont pas synonymes, il est clair (et c'est le cas de façon générale dans le monde anglo-saxon) que ces deux éléments sont liés : le code comme moyen de créer, l'informatique comme moyen de savoir ce que l'on fait.

En France également, le président de la République s'est emparé du phénomène *code* et a rapidement proposé que cela soit enseigné dans les écoles de France. On notera cependant que la liaison avec la science informatique n'a pas, elle, été faite.

4. Qu'est-ce que le numérique ?, Milad Doueihi, Hermès, 2013.

5. Définition du Conseil scientifique de la SIF : <http://binaire.blog.lemonde.fr/files/2014/01/14.Informatique.pdf>

6. <https://www.youtube.com/watch?v=6XvmhE1J9PY>

Le code, pour quoi faire ?

Si politiques et médias semblent s'enthousiasmer devant l'idée de faire enseigner le code, il nous appartient, en tant qu'informaticiens, de rappeler qu'apprendre à programmer n'a jamais été une fin en soi. Cela peut sembler clair quand on écoute Obama, c'est malheureusement moins clair quand on entend certaines déclarations⁷. Si l'on devait aujourd'hui résumer les raisons pour lesquelles il faut l'enseigner, mettons en avant les deux suivantes :

- le code pour la *création numérique* : oui, on peut créer ses jeux, ses applications pour portable, ses expérimentations, ses œuvres numériques,
- le code pour mettre en œuvre la *pensée informatique* : la résolution de problèmes, au XXI^e siècle, cela s'enseigne. Et cet enseignement peut reposer en grande partie sur le traitement informatique.

Il y a également un *effet de bord* envisageable : l'acquisition des compétences numériques mises en avant régulièrement.

Nous renvoyons le lecteur sur d'autres analyses, publiées dans 1024 ou ailleurs, pour retrouver les raisons pour lesquelles un enseignement de l'informatique (et donc du code) s'avère aujourd'hui indispensable !

Nous illustrerons cependant le chemin à parcourir en citant cet extrait du programme national d'informatique d'un pays voisin⁸ :

Le programme national pour l'informatique a pour but de s'assurer que tous les élèves :

- *peuvent comprendre et appliquer les principes fondamentaux et concepts de l'informatique, ce qui inclue l'abstraction, la logique, l'algorithmique et la représentation des données ;*
- *peuvent analyser les problèmes de façon informatique et ont une expérience pratique répétée de l'écriture de programmes informatiques permettant de résoudre de tels problèmes ;*
- *peuvent évaluer et appliquer les technologies de l'information incluant des technologies nouvelles et non familières afin de résoudre de façon analytique des problèmes ;*
- *sont des utilisateurs créatifs responsables, compétents, en confiance des technologies de l'information et de la communication.*

7. En particulier, si l'on suppose qu'une formation courte de trois jours peut suffire.

8. http://www.computingatschool.org.uk/data/uploads/primary_national_curriculum_-_computing.pdf

Qu'est-ce que Class'Code ?

Le cahier des charges

En janvier 2015, l'État a lancé, par le Fonds national d'innovation, l'action « Culture de l'innovation et de l'entrepreneuriat »⁹. Celle-ci contenait un volet « code », dont le but était de fournir des solutions permettant de former la jeunesse de France au code.

Un consortium s'est réuni, a demandé à Inria de porter le projet, l'a soumis. Le projet¹⁰ a été retenu en juin 2015.

D'autres projets ont également été retenus par le jury. Un objectif de Class'Code est de travailler avec ces autres projets. Certains sont plus généralistes, d'autres s'attachent à former une classe particulière de la population. L'originalité de Class'Code réside dans le couplage fort entre informatique et code, la mise en avant de certaines valeurs, et le choix de se concentrer sur la formation des éducateurs.

Class'Code en 20 lignes

Ce projet a pour ambition de mettre en place et de déployer sur tout le territoire une formation hybride à destination des professionnels de l'éducation et de toutes les personnes désirant initier les jeunes à la pensée informatique. Il s'agit de former sur 5 ans, en ligne et sur environ 1 500 lieux, les 300 000 formateurs dont le pays a besoin, y compris dans les milieux géographiquement et socialement écartés.

Pour répondre à cet enjeu, nous nous engageons à :

- créer une formation de formateurs ouverte et accessible à toutes et tous,
- inciter les professionnels du numérique à soutenir cet effort de formation en accompagnant cette démarche au fil du temps,
- populariser l'informatique ludique et créative dans la littérature numérique, pour les jeunes qui débutent en sciences informatiques.

En construisant une formation hybride, associant une formation en ligne à un maillage du territoire, nous permettrons ainsi aux éducateurs, animateurs et enseignants désireux d'ajouter la compétence informatique de se former, aux professionnels de l'informatique d'apprendre à transmettre leurs compétences aux enfants, aux mairies et collectivités locales d'identifier les bonnes compétences.

Class'Code doit permettre à quelqu'un de transmettre les aspects tant techniques que sociétaux liés au code, en s'appuyant sur une vision scientifique de celui-ci. La transmission sera adaptée au public visé, reposant sur ses aspects ludiques et créatifs sans masquer le caractère scientifique et technique.

9. <http://www.education.gouv.fr/cid91267/investissements-d-avenir-culture-de-l-innovation-et-de-l-entrepreneuriat.html>

10. Initialement, le projet devait s'appeler MAAISON : Maîtriser et Accompagner l'Apprentissage de l'Informatique pour notre Société Numérique.

Et l'Éducation nationale ?

Une question qui revient systématiquement est celle du *choix* du périscolaire : pourquoi est-ce là que nous portons nos efforts ? Pourquoi ne pas directement nous adresser à l'École de la République ?

— La première raison est positive : nous sommes très heureux de travailler aujourd'hui avec les acteurs de l'éducation populaire ! L'éducation populaire est aujourd'hui l'acteur principal du périscolaire. Associations et fédérations sont présentes un peu partout en France : leurs animateurs interviennent du secteur sportif à celui du périscolaire dont l'importance est grandissante avec la réforme des rythmes scolaires. Il y a une énergie phénoménale qui se dégage quand on discute avec ces partenaires. Mieux, ils sont ceux grâce à qui le maillage du territoire, ingrédient essentiel de Class'Code, va pouvoir avoir lieu.

— La seconde est plus politique : l'initiation à l'informatique a été confiée au périscolaire. Le précédent ministre de l'Éducation nationale l'a indiqué clairement durant l'été 2014, la SIF et l'EPI se sont vu confirmer cela lors de différents entretiens au MEN en 2014 et 2015, les programmes des cycles 2 et 3¹¹ ne laissent pas penser que l'enseignement de l'informatique soit un objectif clair du système scolaire dans le primaire.

Les différents choix de Class'Code rendront cependant possible dans les formations de l'Éducation nationale l'intégration du matériel pédagogique préparé : des partenaires déjà dans Class'Code ont l'habitude de travailler à la formation des enseignants, plusieurs ESPE¹² seront prochainement partenaires de Class'Code et, surtout, l'ensemble du matériel sera distribué sous une licence *Creative Commons* rendant facile sa réappropriation par l'Éducation nationale.

Finalement, seuls manquent les cadres politique, institutionnel et financier permettant le déploiement des ressources et méthodes Class'Code dans la formation initiale et/ou continue des enseignants. C'est la prérogative des pouvoirs publics de proposer ces cadres.

Qui est Class'Code ?

Class'Code a été créé suite à la rencontre d'un certain nombre de personnes, d'entreprises et d'organisations.

— La Société informatique de France milite en faveur d'un enseignement de l'informatique pour tous depuis sa création en 2012.

11. <http://www.education.gouv.fr/cid95812/au-bo-special-du-26-novembre-2015-programmes-d-enseignement-de-l-ecole-elementaire-et-du-college.html>

12. Les Écoles Supérieures du Professorat et de l'Éducation sont les lieux de la formation des enseignants

— Inria, parmi les différents rôles qu'il a aujourd'hui vis-à-vis de l'informatique en France, joue un rôle essentiel en ce qui concerne la médiation scientifique : les sites Interstices (<https://interstices.info/>) et Pixees (<https://pixees.fr/>) sont bien connus des informaticiens, même s'ils sont gérés en partenariat avec d'autres acteurs (dont la SIF).

— Magic Makers (<http://magicmakers.fr/>) est une petite (par sa taille) entreprise qui très vite, avec l'introduction du langage Scratch (<https://scratch.mit.edu/>), a cherché à monter des ateliers en direction des enfants et des adolescents. Leur expérience est vite devenue une référence en France.

— OpenClassrooms (<https://openclassrooms.com/>) est une entreprise dont le but est de monter des MOOCs. Depuis plusieurs années, leur savoir-faire s'est développé au point d'en faire un opérateur de MOOCs de référence.

D'autres partenaires ont vite rejoint cette aventure : des organisations professionnelles de l'informatique ont vite adhéré à cette idée, des associations reconnues de l'éducation populaire ont également choisi de partager notre chemin et ce projet.

Enfin, deux régions ont accepté, dès l'écriture du projet, de piloter avec nous celui-ci, d'investir leur énergie, de mettre à la disposition de Class'Code personnels et salles : la région Pays de la Loire, et la région Provence-Côte d'Azur. Avec elles, deux universités (Nice et Nantes) sont venues renforcer le premier cercle.

Il convient de remercier en particulier la région Pays de la Loire qui a soutenu le projet dès le début. Mieux, l'amorçage financier qu'ils ont consenti a permis au projet de démarrer avant sa signature par la Caisse des dépôts. Ce sont ainsi 6 mois qui ont été gagnés !

Enfin, rappelons que le projet est soutenu par le deuxième programme Investissements d'avenir (PIA 2), avec pour opérateur la Caisse des dépôts.

Class'Code : les valeurs

Au-delà du gigantisme de la tâche, il est intéressant de préciser quels sont les défis réels auxquels Class'Code s'attaque. Ces défis, nous les avons transformés en principes, c'est-à-dire en éléments sur lesquels nous serons en continuelle vigilance.

Trois principes gouvernent notre action :

- le code comme point d'entrée vers la science (informatique), la technique et la société,
- l'équité territoriale,
- une démarche prenant en compte les enjeux de genre.

Trois choix sont au service de ces principes :

- le soutien des informaticiens (étudiants, élèves, techniciens, ingénieurs),
- la création de ressources éducatives libres,
- la recherche.

Rentrons maintenant dans les détails.

Les principes

L'informatique au cœur du numérique

La confusion générale (voir le paragraphe « Mais au fond, que veut-on enseigner ? », page 16) concernant les liens (ou non) entre l'informatique et le numérique, l'emploi d'un mot (code) qui n'a pas de définition académique stabilisée, permet d'envisager la formation au code de multiples façons.

Comprendre que son enseignement pouvait être très ludique est également de nature à laisser penser qu'une approche reposant simplement sur le *savoir reproduire* pourrait être suffisante.

Le projet Class'Code propose une autre vision du code. Celle-ci est également celle défendue dans le rapport de l'Académie des sciences¹³.

L'équité territoriale

Une difficulté identifiée est la taille et la complexité du territoire à atteindre : Paris et province, villes et campagnes, quartiers de très différentes natures, lieux sur lesquels on peut compter sur des acteurs de terrain très expérimentés et motivés ou pas encore...

L'utilisation dans la formation de FLOTs ou MOOCs (*massive open online courses*) créés spécifiquement dans Class'Code est un premier élément permettant de passer à l'échelle. Mais cette solution réclame un accompagnement pour permettre aux éducateurs et animateurs en formation d'avoir les meilleures chances de suivre complètement celle-ci.

Les MOOCs hybrides semblent être la solution à retenir pour être présents partout et permettre au plus grand nombre de voir aboutir leurs efforts. En anglais, « hybride » se traduit par *blended* ; un enseignement est dit hybride s'il s'appuie sur une ressource externe et sur une part de présentiel. Typiquement, des ressources sont accédées à l'avance par les apprenants d'une classe (ou formation) donnée : ceux-ci ont ensuite un enseignant face à eux qui utilise le matériel pour des activités en présentiel. C'est le principe de la *classe inversée*. Pour l'enseignant universitaire d'hier (et d'aujourd'hui), un point de comparaison est un cours magistral suivi par des travaux dirigés. Simplement, le cours n'est pas fait pas l'enseignant¹⁴.

Le modèle est discutable et perfectible, mais ce qui est complètement différent dans Class'Code est le fait que deux des prérequis principaux seront absents : l'enseignant et la classe. En effet, il n'est pas prévu de *formateurs de formateurs*, et il est essentiel qu'aux regroupements organisés par les associations puissent s'ajouter ceux proposés librement et spontanément par les personnes inscrites au MOOC.

Cela conduit à gérer deux enjeux séparés pour réussir ce MOOC hybride :

13. http://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf

14. Bien entendu, c'est quand même plus compliqué que ça.

- le premier est de nature logistique : à la différence d'une formation continue d'une entreprise (ou de l'Éducation nationale), ici le candidat l'est à un MOOC : il va donc s'inscrire et doit se voir proposé l'accès non seulement aux ressources en ligne, mais aussi aux points de rassemblement. Il faut donc proposer, à l'échelle de la France, des lieux de rassemblement, des horaires, des outils informatiques permettant aux membres de celui-ci d'entrer en contact, de discuter ensemble, de garder le contact après la formation.
- l'autre est pédagogique : il ne s'agit pas de réunir les gens pour que ceux-ci se contentent de discuter des activités vues *online* les jours précédents. L'absence d'enseignant présent pourra être palliée par la présence de facilitateurs, d'apprenants dont les profils font qu'ils savent expliquer les concepts informatiques.

Le genre

Tout enseignant d'informatique, tout cadre de l'industrie, sait qu'aujourd'hui le nombre de filles choisissant d'étudier l'informatique ou d'occuper un emploi dans le secteur numérique est très faible. La bonne foi des uns et des autres n'est pas à mettre en cause : aucun enseignant n'utilise sciemment de critère sexiste dans ses recrutements ou dans ses notations, les entreprises mettent les bouchées doubles pour attirer des candidates. Et malgré les prises de conscience¹⁵, malgré les initiatives, on peine à voir le changement. Néanmoins, d'autres ont montré qu'avec des efforts, la situation pouvait être corrigée¹⁶. Class'Code doit affronter exactement les mêmes questions. Le projet les aborde de la façon suivante :

- l'équipe de Class'Code est volontairement très mixte ;
- nous proposons une démarche proactive : tous les contenus produits sont systématiquement examinés à la loupe de l'égalité des genres ;
- nous maintenons une activité de veille et discutons avec différent-e-s expert-e-s de ces questions.

Les moyens au service de ces principes

L'engagement des partenaires, le professionnalisme des différents acteurs, les relations tissées par les uns et les autres avec des initiatives parallèles en France ou dans d'autres pays... Tout cela concourt au succès de Class'Code mais aussi au respect de nos engagements.

Mais certains aspects spécifiques du projet sont de véritables piliers sur lesquels il nous est possible de défendre les principes que nous venons d'énoncer.

15. Le lecteur de 1024 attend avec impatience le numéro spécial qui suit le congrès de la SIF de 2015 dont le sujet était justement les femmes et l'informatique.

16. On lira par exemple le récent billet : <http://techcrunch.com/2015/11/24/diversifying-technology-education/>.

Le soutien des informaticiens

Une analyse d'expériences dans d'autres pays a montré l'importance d'associer les professionnels de l'informatique et du numérique à des initiatives ambitieuses visant à éduquer toute une classe d'âge à l'informatique, en particulier dans des situations dans lesquelles le corps enseignant n'a pas au préalable suivi une formation suffisante. Cela a été le cas, par exemple, aux Pays-Bas, en Israël, au Royaume-Uni.

Dès le début, les professionnels de l'informatique et du numérique ont donc été associés à Class'Code. Leur rôle peut être

- d'accompagner des non informaticiens qui apprennent (notion de personnes ressources),
- de recruter des informaticien-ne-s qui ont envie d'accompagner les non informaticiens,
- d'accompagner des non informaticiens qui se retrouveront face aux enfants,
- de convaincre les décisionnaires locaux,
- de contribuer à recruter ces personnes,
- de communiquer et aider à communiquer,
- d'aider à valoriser l'investissement dans Class'Code : comment les entreprises, les universités, les écoles d'ingénieurs peuvent-elles soutenir les gens qui s'investissent ?

Le chantier est encore ouvert : des discussions avec les collectifs représentant les professionnels de l'informatique et du numérique (dont la SIF) doivent émerger des idées...

Les ressources éducatives libres

Les ressources éducatives libres (REL) sont des *communs* : l'idée est tout simplement que les ressources éducatives préparées par un enseignant ou un groupe d'enseignants puissent resservir au plus grand nombre. Au-delà d'un principe de gratuité, pour qu'une ressource soit libre, on demande qu'elle respecte la règle des 5 « R »¹⁷ :

- *Retain* : le droit de prendre la ressource, la stocker, la dupliquer,
- *Reuse* : le droit d'utiliser ces ressources en particulier dans ses cours, mais aussi sur un site web, à l'intérieur d'une vidéo,
- *Revise* : le droit d'adapter la ressource ou le contenu (en particulier le droit de traduction),
- *Remix* : le droit de créer une nouvelle ressource en mélangeant des morceaux de ressources existantes,
- *Redistribute* : le droit de distribuer des copies du matériel original, le matériel modifié, le matériel remixé.

17. <http://www.opencontent.org/definition/>

Ces droits se traduisent le plus souvent par une licence *Creative Commons*, sous une forme ou une autre, la plus libre étant la « CC BY » demandant essentiellement que soit respecté le droit de citation : la nouvelle ressource doit citer l'auteur de la première ressource.

Le choix effectué clairement par Class'Code dès l'initiation du projet de mettre à disposition de cette façon les ressources créées n'obéit pas simplement à un affichage politique. Il est motivé par le soutien que différentes organisations apportent à ce type de démarche et à l'ambition du projet de jouer un rôle important :

- L'OCDE soutient la création de REL : on retrouve son argumentaire dans différents rapports¹⁸ qui expliquent que les systèmes d'éducation ont vraiment besoin de beaucoup plus s'appuyer sur les expériences, les idées innovantes des enseignants eux-mêmes. Les REL offrent cette possibilité.
- La « déclaration de Paris¹⁹ » marque l'importance accordée par l'Unesco aux REL.

Le choix effectué, au-delà de se justifier dans un contexte international, permettent d'envisager un succès de Class'Code au-delà de notre capacité de maillage : même dans des endroits où aucune association partenaire n'est présente, il est simple de se rassembler et d'utiliser directement les ressources Class'code pour se former.

Enfin, ce choix est consistant avec la volonté dans Class'Code de réutiliser des ressources produites par d'autres, par exemple Pixees ou Interstices.

La recherche

Plusieurs des problèmes à régler dans Class'Code n'ont pas de réponse immédiate. Ces problèmes auront besoin de solutions technologiques qui pourront donner lieu à une recherche-action spécifique.

- Une sorte de classe inversée sans professeur et sans classe... tel est le défi pédagogique qu'il s'agit de relever.
- Si l'inscription à un MOOC est une chose connue, la capacité de proposer des outils semi-automatiques de création de regroupements à la demande ne l'est pas. Proposer des solutions pour cela est un véritable challenge.
- Le MOOC va générer quantité de données, de traces d'apprentissage ; celles-ci viendront des activités en ligne et des regroupements : une observation de ces traces, une analyse poussée, doivent permettre l'amélioration de la formation, mais aussi plus largement une observation fine du modèle de MOOC hybride que nous proposons.

18. Report Open Educational resources : a Catalyst for Innovation http://www.oecd-ilibrary.org/education/open-educational-resources_9789264247543-en

19. <http://fr.unesco.org/news/la-mise-en-%C5%93uvre-de-la-d%C3%A9claration-de-paris-sur-les-ressources-%C3%A9ducatives-libres-lancement-du>

— Dans le domaine des Sciences de l'éducation également, des questions se posent : Class'Code est un chantier éducatif qui mérite d'être observé et analysé.

De façon plus prospective et générale, nous espérons que dans et autour de Class'Code la recherche trouve toute sa place.

Class'Code, pratiquement

Le projet Class'Code a été lancé officiellement le 15 septembre 2015. Depuis s'organisent tous les aspects du projet :

— La construction pédagogique réunit, sous la direction de Magic Makers, des spécialistes de la formation au code, des informaticiens, des ingénieurs pédagogiques : leur but est de construire les cinq MOOCs qui seront l'armature de la formation. Chaque MOOC est indépendant, deux ont un rôle spécial. Le premier concerne l'activité de codage à proprement parler : ne pas le suivre hypothèque tout de même sérieusement l'intérêt des autres. Un autre MOOC concerne les aspects *projet* dans le code : comment transmettre cette approche de gestion de projet aux jeunes ? Les trois autres MOOCs porteront sur les questions de données/information, de robotique, de réseau/web. Comment construire ses propres éléments d'enseignement avec cette approche ? La construction pédagogique concerne également les activités effectuées lors des rassemblements : quels « kits » proposer ? Comment permettre leur appropriation même (et surtout) quand il n'y a pas de formateur avancé présent ? Tels sont quelques-uns des défis à relever.

— Le *maillage* du territoire consiste à permettre d'identifier tous les acteurs motivés par cette question : il y a en France de nombreux autres collectifs dont le but est de former au code. Class'Code n'a pas vocation à fédérer directement cette communauté, mais on peut espérer que les objets partagés conçus par Class'Code le permettront.

— Le projet soulève des questions non triviales : peut-on réussir un MOOC hybride ? Peut-on fonctionner en ressources éducatives libres ? Peut-on tenir un discours ne réclamant pas des moyens technologiques toujours plus gourmands ? Ces questions tiennent de la politique et se résolvent par le dialogue, par la discussion entre partenaires du projet mais aussi avec tous les autres acteurs de la société concernés. Ces questions se résolvent également par l'invention de nouveaux processus, par la création de nouvelles technologies, par la recherche.

Concrètement, les équipes de Class'Code entendent proposer à tous le premier MOOC à l'automne 2016. Avant cela, des expérimentations « pilotes » auront lieu, permettant de vérifier le dispositif. D'ores et déjà, la préinscription est possible sur <https://classcode.fr/>.

La SIF, Class'Code a besoin de vous

Le succès de Class'Code reposera sur de nombreux facteurs : la qualité du matériel préparé, l'importance de l'enjeu pour les collectivités locales (en particulier les mairies qui auront à financer le périscolaire), l'envie d'apprendre l'informatique des animateurs et éducateurs. Mais aussi l'investissement des professionnels du numérique et de l'informatique.

Il existe de nombreuses façons pour ceux-ci d'aider Class'Code (et bien entendu les membres de la Société informatique de France sont en première ligne !) :

- Les MOOCs en préparation sont hybrides et seront accompagnés de regroupements ; lors de ceux-ci des questions seront posées, des non-informaticiens pourront avoir des difficultés avec la mise en pratique, les concepts, l'envie d'aller plus loin. Le grand espoir (et choix) de Class'Code est que des informaticiens pourront être présents durant ces regroupements pour *faciliter* l'accès au matériel, accompagner l'apprentissage des non-informaticiens.
- Pour ces informaticiens, il y a sans doute aussi des choses à apprendre, à redécouvrir, à explorer : les langages de programmation comme Scratch, des manipulations de robots, l'histoire de l'informatique. Mais aussi, et surtout, comment enseigner cela à des enfants de 8 à 12 ans ?
- On le voit régulièrement dans les médias : le code serait facile à enseigner et ne demanderait aucune compétence, aucune formation... les raisons de l'enseigner sont elles aussi insuffisamment expliquées... Il faut donc intervenir dans le débat public pour expliquer, faire comprendre l'intérêt dans ce contexte de Class'Code. Pour que, demain, les équipes municipales proposent Class'Code à leurs employés, choisissent des animateurs ayant suivi Class'Code, il est nécessaire de relayer les messages.
- Des collègues, des étudiants, des élèves vont choisir de suivre Class'Code, de faciliter l'accès à l'informatique, peut-être aussi d'aller dans les écoles et s'adresser directement aux enfants. Pour cela, ils seront allés s'inscrire sur <https://classcode.fr>. En tant que membres de la SIF, soyons tous réactifs à notre niveau. Nous pouvons aussi être des facilitateurs... en trouvant des moyens pour que nos étudiants intègrent ces activités dans leur cursus, en mettant en avant les bons exemples, en offrant la parole à celles et ceux qui promeuvent ces activités.

La SIF est, avec d'autres, à l'origine de Class'Code. Elle a de ce fait une responsabilité particulière : celle de contribuer à son succès.



Comment simuler efficacement les interactions lumière/matière en synthèse d'image

Adrien Gruson et Kadi Bouatouch¹

La synthèse d'image a connu une évolution fantastique ces deux dernières décennies et son champ d'application s'est fortement élargi. Toutes les méthodes et tous les algorithmes dérivés de la synthèse d'image sont maintenant utilisés dans des domaines autres que celui de l'image tels que l'infrarouge, le son, la téléphonie, etc. Le réalisme des images de synthèse ne cesse de croître grâce aux nombreuses méthodes sophistiquées de rendu capables de simuler les diverses interactions lumière-matière comme : la réflexion, la réfraction, les diffusions sous-surfaciques ou dans un milieu semi-transparent (appelé aussi milieu participant). Ces différentes interactions sont prises en compte dans divers chemins de lumière qui connectent les sources de lumière à la caméra (ou observateur). La majorité des méthodes de rendu, utilisées dans le domaine du jeu vidéo et du cinéma, sont basées sur une approche stochastique comme la méthode de Monte Carlo. Cet article met l'accent sur cette classe de méthodes permettant la création d'image toujours plus proche de la réalité.

Principes et définitions

La synthèse d'image a connu une évolution significative durant ces dernières décennies. En effet, il y a peu de temps c'était un ensemble de techniques permettant de produire différents effets (perspective, ombres, transparence, textures...) plus ou moins fidèlement. Elle s'appuyait uniquement sur des modèles empiriques dont le seul but était d'obtenir une représentation de ces effets sans chercher à les reproduire fidèlement. Le but recherché était alors de produire des images *réalistes*.

1. IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. E-mail : adrien.gruson@irisa.fr, kadi.bouatouch@irisa.fr.

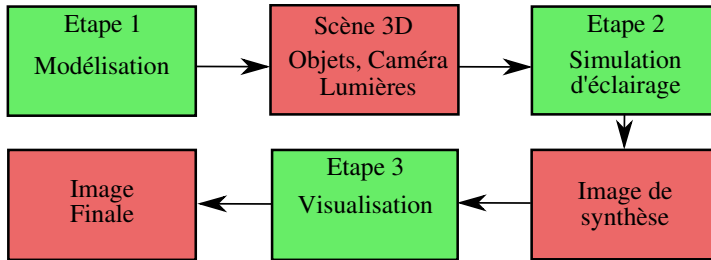


FIGURE 1. Différentes étapes (boîtes vertes) de la synthèse d'image. Chaque étape produit une donnée (image ou scène, boîtes rouges) qui est ensuite utilisée dans l'étape suivante.

Maintenant, la synthèse d'image a pris une direction plus scientifique en s'appuyant sur différents domaines de la science tels que la perception visuelle, la physique et les mathématiques. L'objectif actuel est de reproduire le plus précisément possible la réalité. Il s'agit donc de produire des images *photo-réalistes* où un utilisateur ne pourrait pas distinguer ce que produit un ordinateur de la réalité. Le processus de création d'une image de synthèse comprend trois parties (Figure 1) : la modélisation géométrique, la simulation des interactions lumière-matière (ou technique de rendu ou simulation d'éclairage) et la visualisation.

La modélisation géométrique consiste d'une part à créer des objets et d'autre part à les placer dans l'environnement 3D. Pour chaque objet, il faut définir ses paramètres géométriques (liste de sommets pour un polygone, normale...) et ses paramètres d'aspect (réflectivité, transparence, textures, couleurs...). Par ailleurs, l'utilisateur place dans la scène des sources de lumière et une caméra virtuelle. Cette caméra virtuelle, qui représente l'observateur, est associée à un écran virtuel (grille de petites surfaces carrées appelées pixels à travers lesquels sont déterminés les objets visibles). Elle a pour but de fixer une direction de vue formée par l'observateur et un point de visée (Figure 2).

Après avoir finement défini cet environnement (ou scène) 3D, la simulation de l'éclairage est réalisée pour produire l'image de synthèse. Le but de cette étape est de calculer la répartition de l'énergie lumineuse reçue par l'observateur à travers chaque pixel de l'écran virtuel. Cette énergie lumineuse est une grandeur photométrique appelée luminance et qui est associée à une distribution spectrale $L(\lambda)$, où λ est la longueur d'onde. L'image de synthèse obtenue est alors représentée par un tableau à deux dimensions de luminances. Pendant cette étape, l'ordinateur calcule toutes les interactions entre les sources de lumière et les objets de la scène. Ces interactions sont multiples (réflexion, réfraction, diffusion) et difficiles à évaluer pour l'ordinateur. C'est pour cette raison qu'il faut parfois plusieurs minutes voire des heures pour générer une image de bonne qualité.

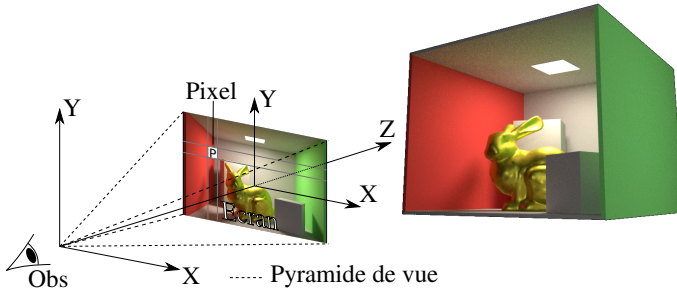


FIGURE 2. Repère de l'observateur et pyramide de vue (caméra)

Enfin, il nous faut passer par l'étape de visualisation. Cette étape va consister à convertir l'image physique en RVB (rouge, vert, bleu). Le but étant de pouvoir afficher l'image de synthèse sur un écran classique d'ordinateur.

Dans cet article nous ne parlerons que de la génération d'images de synthèse et pas de séquence vidéo. Cependant, pour générer une vidéo, il faut décrire le mouvement des objets de la scène, de la caméra, des sources de lumière et les changements d'aspect de ces objets dans le temps. Tout objet mobile est caractérisé par une trajectoire spatio-temporelle et des déformations plus ou moins complexes. Ensuite, cette scène animée est envoyée au logiciel de simulation d'éclairage qui va décomposer cette animation en une séquence d'images de synthèse (50 par seconde).

Dans la suite de cet article, nous présentons les différents types d'interaction lumière-matière qui sont pris en compte dans l'équation de luminance (aussi appelée l'équation de rendu). Puis, nous présentons les méthodes stochastiques de rendu permettant de résoudre l'équation de luminance afin de produire une image de synthèse photo-réaliste. Enfin, nous présentons l'étape de visualisation de ces images.

Simulation d'éclairage ou technique de rendu

Une source de lumière est considérée comme un rayonnement électromagnétique défini par un spectre, représentant une énergie lumineuse qui est une fonction de la longueur d'onde. La lumière émise par une source lumineuse interagit avec les objets d'un environnement, elle est soit réfléchie, soit réfractée, soit diffusée par un milieu participant (fumée, poussière, feu, brouillard, nuage).

Pour atteindre le photo-réalisme, les efforts de recherche ont porté principalement sur l'élaboration de modèles d'éclairage permettant de rendre ces effets. L'équation de rendu [3] modélise ces effets :

$$L_r(\vec{x}, \omega_r) = \int_{\Omega} f_r(\vec{x}, \omega_r, \omega_i) \cdot L_i(\vec{x}, \omega_i) \cdot |\omega_i \cdot N| \cdot d\omega_i$$

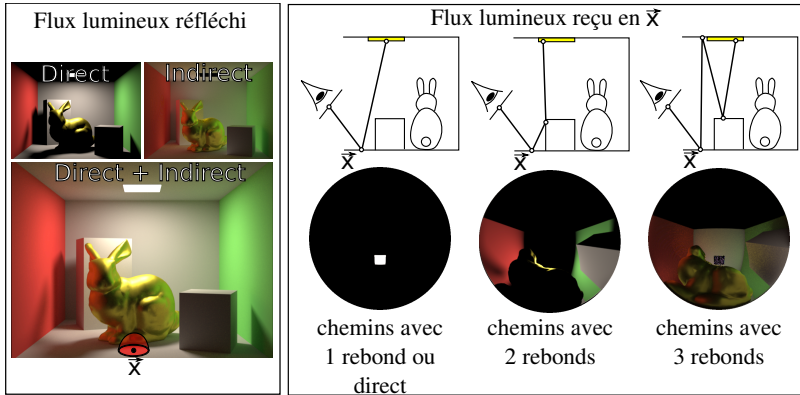


FIGURE 3. Les différents types de chemins lumineux. À gauche : les images dues à l'éclairage direct, indirect et la combinaison des deux. À droite : un aperçu de la luminance incidente au point \vec{x} pour différentes longueurs de chemin (nombre de rebonds).

où L_r est la luminance réfléchie au point \vec{x} vers l'observateur représenté par la direction ω_r . Cette luminance réfléchie est égale à l'intégrale sur l'ensemble Ω des directions incidentes ω_i au point \vec{x} . Pour chaque direction d'incidence ω_i , il faut connaître la luminance incidente L_i et le matériau au point \vec{x} . En effet, lors d'une interaction lumière-matière, seulement une certaine quantité de cette luminance incidente est réfléchie/réfractée vers l'observateur et dépend du matériau (décrit par le terme $f_r(\vec{x}, \omega_r, \omega_i)$). $|\omega_i \cdot N|$ est le cosinus de l'angle d'incidence entre ω_i et la normale N au point \vec{x} .

En ce qui concerne la luminance incidente en \vec{x} , elle est la conséquence de plusieurs facteurs qui sont résumés dans la figure 3. Premièrement, les sources de lumière peuvent éclairer directement \vec{x} . Si on ne prend en compte que cette contribution lumineuse et que l'on calcule le flux réfléchi en tout point visible de la scène, cela nous donne une image due à l'éclairage *direct* (Figure 3). Deuxièmement, il faut prendre en compte les contributions des autres objets de la scène (aussi appelées éclairage indirect). Ces contributions correspondent au fait que tous les objets sont éclairés par les sources de lumière et réfléchissent vers \vec{x} . On parle de chemin de lumière indirect généré après plusieurs rebonds de lumière (Figure 3 avec 2 et 3 rebonds). Pour générer une image de synthèse, il faut combiner ces deux contributions (indirect et direct).

En ce qui concerne les matériaux, ils permettent de modéliser l'aspect de l'objet. Par exemple, le fait que l'objet soit rouge en dessous d'un éclairage blanc est dû à son matériau. Par ailleurs, un matériau définit la proportion de lumière réfléchie vers

l'observateur. Il existe plusieurs modèles modélisant cette proportion, en voici les principaux :

— **Réflexion diffuse**

Une lumière incidente arrivant sur un objet mat pénètre dans celui-ci puis ressort avec la même amplitude quelle que soit la direction d'observation (de réflexion), c'est la réflexion diffuse. C'est par exemple le cas des murs de couleur dans la scène de la figure 3.

— **Réflexion spéculaire (objets brillants)**

Une lumière incidente arrivant sur un objet lisse (réfléchissant) est réfléchi par la surface de celui-ci mais uniquement dans certaines directions autour de la direction de réflexion idéale. Cette dernière est telle que l'angle de réflexion est égale à l'angle d'incidence. Par exemple, un miroir est quasiment parfaitement spéculaire, il n'y a qu'une seule direction de réflexion spéculaire qui est celle de la réflexion idéale. Dans la scène de la figure 3, le lapin est constitué d'un matériau brillant.

En résumé, le matériau, associé à un objet, caractérise la capacité de l'objet à réfléchir, réfracter et diffuser la lumière.

Méthodes de rendu stochastiques

Nous avons vu précédemment que les effets lumineux sont modélisés par l'équation de rendu. Cependant, nous avons vu que la luminance incidente est la contribution de chemins lumineux de différentes longueurs (ou nombre de rebonds). Une façon de considérer le problème de la simulation d'éclairage est de trouver l'ensemble des chemins lumineux qui contribuent à la luminance des pixels de l'écran de la caméra virtuelle. Plus formellement, on peut définir un chemin lumineux comme $\bar{x} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n)$ où \bar{x}_i est le $i^{\text{ème}}$ sommet (point d'intersection entre un rayon lumineux et un objet de la scène) composant ce chemin. La luminance I_j d'un pixel j (de l'écran) est égale à :

$$I_j = \int_P f_j(\bar{x}) d\mu(\bar{x}),$$

où P est l'ensemble des chemins lumineux et f_j est la fonction de contribution du chemin \bar{x} au pixel j . Cette fonction prend en compte les différentes valeurs des matériaux des objets avec lesquels le chemin lumineux interagit (rebond) et l'énergie émise par la source de lumière. Cette fonction peut être égale à zéro si le chemin n'apporte pas d'énergie lumineuse au pixel p .

En informatique, on sait évaluer la valeur d'une intégrale de grande dimension. Cependant, plus l'espace d'intégration est grand, plus son évaluation est coûteuse. Le problème est, qu'en synthèse d'image, nous avons affaire à des intégrales de très grande dimension. Monte Carlo est une technique qui permet d'approcher la valeur de l'intégrale (l'ensemble des chemins) en se basant uniquement sur un échantillonnage de l'espace des chemins. L'idée est de ne pas considérer tous les chemins de

lumière possibles mais uniquement prendre un échantillon de chemins possibles. Si l'échantillon est assez grand (beaucoup de chemins) et représentatif, alors on peut l'utiliser en considérant qu'il représente la totalité des chemins lumineux. Dans une approche Monte Carlo, on parle d'estimateur et celui-ci a la forme suivante :

$$E[I_j] = \frac{1}{N} \sum_{i=1}^N f_j(\bar{x}_i) / p(\bar{x}_i),$$

où N est le nombre de chemins utilisés dans l'échantillon et \bar{x}_i un chemin lumineux construit de manière aléatoire. La densité de probabilité $p(\bar{x}_i)$ permet de construire les chemins lumineux. En effet, l'idée est de construire des chemins qui contribuent à l'image finale et d'éviter les chemins non contributifs. La précision de l'estimateur (son erreur) va dépendre du nombre d'échantillons (N) utilisés et de la qualité des chemins générés ($p(\bar{x}_i)$). Augmenter le nombre d'échantillons fait croître de façon linéaire le temps de calcul. Donc, la seule façon d'avoir un calcul plus efficace est de recourir à de bonnes méthodes de construction des chemins lumineux.

Pour résumer, le but de la simulation d'éclairage est de construire des chemins lumineux contributifs. Ces chemins sont construits par morceaux en lançant des rayons. Un rayon est défini par sa position dans l'espace 3D et sa direction. Un des modules basique du moteur de rendu est celui qui calcule l'intersection de ce rayon avec les objets 3D de la scène. Ci-après, nous présentons différentes façons de construire des chemins lumineux.

Chemins générés à partir de la caméra

Une première possibilité est de générer un chemin lumineux à partir de la caméra (Figure 4). Pour commencer, on lance un rayon à partir d'un observateur à travers un pixel de l'écran virtuel vers les objets de la scène. À chaque intersection (point \bar{x} dans la figure 3), on veut calculer la contribution des sources de lumière. Pour ce faire, on va choisir de façon aléatoire un point sur une des sources de lumière et on lance un rayon dans sa direction. Si le rayon arrive à destination, alors on sait que le point est visible et on ajoute sa contribution. Cependant, cette approche permet de construire uniquement des chemins *directs*. Pour des chemins plus longs (indirects de longueur N comme dans le figure 3), nous devons étendre le chemin actuel. Pour ce faire, nous lançons un nouveau rayon depuis le dernier point d'intersection et dans une direction aléatoire. Ce nouveau rayon génère un nouveau point d'intersection et on recommence ce processus à partir de ce nouveau point (sélection d'un point sur une source de lumière, direction de rebond choisie de façon aléatoire). Pour éviter de construire des chemins de longueur infinie, avant de faire rebondir un rayon, un test aléatoire permet de décider de l'arrêt ou non de ce processus.

Chemins générés à partir d'une source de lumière

Une seconde possibilité est de générer un chemin lumineux à partir d'une source lumineuse (Figure 4). On commence par échantillonner, de manière aléatoire, un

point sur une source de lumière. Puis, un rayon est tracé dans une direction aléatoire. En chaque intersection, on essaye de déterminer si ce point est visible par l'observateur. Pour ce faire, on génère un rayon partant de notre point d'intersection actuel vers l'observateur. Si ce rayon touche l'écran de la caméra, alors on ajoute la contribution de ce chemin lumineux au pixel impacté.

Discussion

Nous avons vu deux façons de construire des chemins lumineux. En synthèse d'image, il existe une multitude d'algorithmes pour construire ces chemins. L'un des plus avancés est le tracé de chemin bidirectionnel où l'on trace deux chemins lumineux en même temps : un chemin depuis la caméra $\bar{x} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n)$ et un depuis la source de lumière $\bar{y} = (\bar{y}_0, \bar{y}_1, \dots, \bar{y}_n)$. Ensuite on essaye de connecter les sommets de ces deux chemins deux à deux. Pour chaque couple de sommets, on génère un rayon pour tester si les deux sommets sont mutuellement visibles ou non. S'ils sont visibles alors nous avons construit un chemin lumineux contributif. La figure 4 résume les différentes approches discutées et les images générées par ces différents algorithmes pour un temps donné (pour comparer l'efficacité de l'un par rapport à l'autre). Par ailleurs, dans cette figure, les images de synthèse sont générées avec différentes techniques de rendu. Ces images nous permettent de juger des différentes techniques : moins il y aura d'artefacts visuels (bruit, objets noirs), plus la méthode de rendu sera jugée efficace.

Metropolis-Hasting

Dans les approches vues précédemment, quelle que soit la technique de rendu choisie, la construction des chemins repose sur des décisions aléatoires (sélection d'un point sur une source de lumière, d'une direction de rebond). Dans la pratique, pour chaque chemin lumineux, un ensemble de nombres aléatoires est généré puis utilisé par la méthode de rendu pour construire un chemin lumineux. Cet ensemble contient des nombres aléatoires indépendants et, à chaque nouveau chemin lumineux, un nouvel ensemble est créé. Cela signifie que la génération d'un chemin n'est pas basée sur les chemins précédemment créés.

Pendant, dans des scènes complexes (lampes extérieures et observateur à l'intérieur d'un bâtiment, de multiples rebonds de lumière nécessaires), il est difficile de générer un chemin contributif. Donc, si par chance, nous réussissons à générer un chemin contributif (appelé aussi *chemin courant*), alors nous voulons l'utiliser pour explorer *localement* l'espace des chemins. Tout ceci dans l'espoir de trouver de nouveaux chemins contributifs proches de ce chemin. L'algorithme de Metropolis-Hasting [2, 6] nous permet cela en générant une marche aléatoire dans l'espace des chemins. À chaque étape de cette marche, il génère un chemin *proposé* \bar{y} en appliquant une mutation (petit changement) aléatoire au chemin \bar{x}_{i-1} qui est notre chemin courant. Si ce nouveau chemin \bar{y} contribue à l'image, il est alors accepté (après calcul

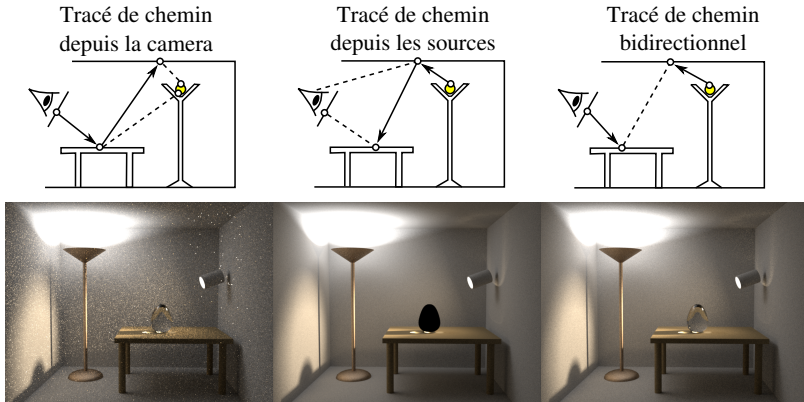


FIGURE 4. Cette image résume les différentes techniques décrites dans cet article. Chaque technique construit différemment les chemins lumineux et génère une image de synthèse (même temps de génération pour ces trois images). Ces différences impactent l'efficacité des algorithmes pour gérer tel ou tel phénomène lumineux. Ce manque est visible dans les images finales sous forme de bruit ou d'effets ignorés (objet noir dans la technique du centre).

de probabilité d'acceptation) et devient le nouveau chemin courant. Sinon, le chemin courant \bar{x}_i est égal à \bar{x}_{i-1} (donc pas de changement de chemin courant) et une autre mutation aléatoire est appliquée à \bar{x}_i .

À la fin de cet algorithme, nous obtenons une séquence de chemins $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_N$ où chaque chemin dépend du chemin précédent. Ceci diffère des approches classiques basées sur Monte Carlo où les chemins sont indépendants entre eux. Enfin, pour générer l'image de synthèse, on utilise cette séquence de chemins dans un estimateur de Monte Carlo.

La scène représentée par la figure 5 permet de voir l'amélioration qu'apportent les algorithmes de rendu basés sur Metropolis-Hasting. Dans cette scène, il est difficile de trouver les chemins de lumière passant par la fenêtre. L'algorithme de Metropolis permet d'explorer localement cette fenêtre avec son mécanisme de mutation. Ceci impacte fortement l'efficacité de l'algorithme de rendu et produit des images sans bruit, ce qui n'est pas le cas des techniques de rendu n'utilisant pas Metropolis (appelées dans cette figure *uniforme*). La figure 6 montre d'autres exemples de rendu basés sur Metropolis-Hasting.



FIGURE 5. Scène Vitraux : éclairage à travers les fenêtres subissant une absorption et une diffusion par la poussière. Pour cette scène, l’algorithme de Metropolis a été utilisé. En effet, il est difficile de trouver les chemins lumineux passant par la fenêtre. Par ailleurs, nous montrons des zooms pour comparer la technique avec et sans (appelée uniforme) Metropolis. On observe une nette amélioration due à l’exploration locale faite par Metropolis.

Visualisation

Comme il a été mentionné précédemment, le processus de synthèse d’image fournit, pour chaque pixel, la luminance d’un point de la scène vue par l’observateur à travers ce pixel. Cette luminance dépend de la longueur d’onde (notons que dans le cas d’une approche trichromatique, trois longueurs d’onde sont utilisées : Rouge, Vert, Bleu). Appelons cette luminance $L(\lambda)$, λ étant la longueur d’onde. Rappelons que le spectre visible correspond à l’intervalle des longueurs d’onde [380 nm, 780 nm]. Pour pouvoir visualiser ces luminances sur un écran vidéo il va falloir les convertir en trois composantes Rouge, Vert, Bleu (RVB). Cette conversion est donnée par :

$$R = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{r}(\lambda) d\lambda$$

$$V = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{v}(\lambda) d\lambda$$

$$B = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{b}(\lambda) d\lambda$$

Les fonctions $\bar{r}(\lambda)$, $\bar{v}(\lambda)$, $\bar{b}(\lambda)$ sont appelées fonctions de mélange trichromatiques. Elles dépendent du moniteur utilisé.

Pour pouvoir à la fois calculer précisément les luminances et les couleurs RVB associées, il est important de considérer un nombre de longueurs d’onde suffisant et

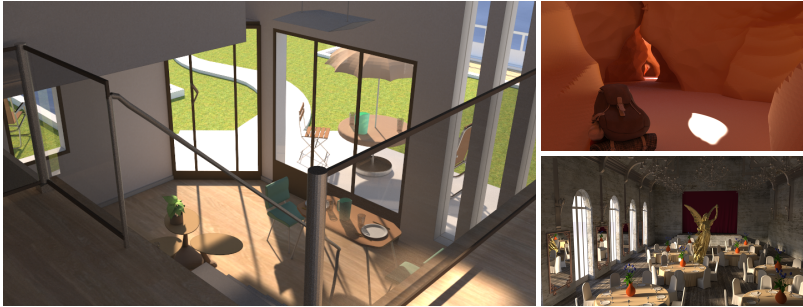


FIGURE 6. D'autres exemples d'images de synthèse générées avec un algorithme basé sur Metropolis.

des valeurs de longueur d'onde adéquates tel qu'il est montré dans les travaux de Meyer [4]. Notons qu'en synthèse d'image on utilise en général uniquement trois longueurs d'onde (Rouge, Vert, Bleu), ce qui évite cette conversion.

Pour les moniteurs classiques, chaque composante R, V, B est codée sur 8 bits (valeur comprise entre 0 et 256). Cependant, en synthèse d'image, nous n'avons pas cette limitation et effectuons la simulation avec une meilleure précision. D'autre part, nous n'avons pas de restriction sur la dynamique (rapport entre la valeur la plus basse par rapport à la valeur la plus haute) de l'image. Cependant, la luminance maximale affichée sur un moniteur est de l'ordre de 100 à 300 candelas par mètre carré alors que la luminance du ciel peut être de l'ordre de quelques milliers de candelas par mètre carré. Ce qui pose des problèmes de restitution de l'image calculée. La solution la plus simple, adoptée couramment, est de tronquer les valeurs ou d'effectuer une mise à l'échelle. Cette solution donne des résultats très insuffisants étant donné la perte d'information induite. Il est donc important de comprimer les valeurs trichromatiques RVB calculées en prenant en compte les propriétés du système visuel humain et du moniteur utilisé. Cette compression est appelée *Tone Mapping* [1, 5].

Conclusion

La synthèse d'image est passée d'un stade empirique à celui des outils et méthodes rigoureux utilisés dans diverses branches de la science. En effet, elle s'appuie sur l'informatique, la physique, la perception visuelle, la photométrie, la colorimétrie et les mathématiques. Autant de domaines qui ne peuvent que témoigner de l'intérêt et de la richesse scientifique de ce merveilleux domaine de recherche qu'est la synthèse d'image.

L'objectif majeur est de reproduire fidèlement et efficacement l'interaction entre la lumière, émise par des sources lumineuses, et les objets composant une scène 3D. Fidèlement veut dire bien simuler les lois de la physique et surtout prendre en compte

les caractéristiques du système visuel humain. Reproduire efficacement consiste à calculer la luminance d'un pixel de manière précise et avec un coût de calcul raisonnable. Ces deux exigences sont maintenant possibles grâce aux algorithmes basés sur les méthodes de Monte Carlo et de Metropolis-Hasting.

Références

- [1] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging : theory and practice*. CRC Press, 2011.
- [2] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1) :97–109, 1970.
- [3] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.
- [4] Gary W Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41(1) :57–79, 1988.
- [5] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging : acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.
- [6] Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997.



Le système LSE

Jacques Baudé¹

Qui a connu ou qui se souvient du LSE qui a largement permis le travail des pionniers de « l'informatique pédagogique » dès les années 70 du siècle dernier ? Cet article a pour objet de replacer le système dans son contexte historique : expliquer sa genèse, montrer ce qu'il a permis au sein d'une communauté enseignante, comprendre les raisons de sa disparition après plusieurs années de bons et loyaux services...

Les collègues intéressés se reporteront aux ouvrages encore disponibles (cf. bibliographie) et aux articles en ligne, notamment sur le site de l'EPI².

Les origines. Pourquoi le LSE ?

Fin 1967 se posait au service informatique de l'École supérieure d'électricité (Supélec) le problème suivant : donner aux élèves de l'école un accès convivial aux ordinateurs de l'époque. Il fut décidé de réaliser un système en temps partagé. Ce fut l'origine de la saga des LSx (LSD, LSE, LSG, LST) développée durant les années 1968-1976³.

Or, à la suite du séminaire CERI-OCDE de Sèvres en mars 1970⁴, il fut décidé – dans le contexte favorable du Plan Calcul, de la préparation du VI^e plan et de

1. Secrétaire général puis président de l'EPI de 1981 à 1995., président d'honneur de l'EPI, jacquesbaude@free.fr.

2. <http://epi.asso.fr>

3. Yves Noyelle, *La Saga du LSE et de sa famille (LSD-LSG-LST)*, <https://edutice.archives-ouvertes.fr/edutice-00001092/document>

4. Jacques Baudé, *Quelques points de repère dans une histoire de 40 ans : L'association Enseignement Public et Informatique (EPI) de février 1971 à février 2011*, http://www.epi.asso.fr/revue/histo/h11epi_jb.htm

« l'impératif industriel » – de lancer l'expérience des « 58 lycées »⁵. La Délégation à l'informatique passa contrat avec Supélec pour la définition d'un langage adapté aux besoins de l'enseignement. Ce fut le LSD, rebaptisé LSE pour des raisons évidentes !

Le rapport de définition du LSE fut remis au ministère de l'Éducation nationale (mission à l'informatique) en octobre 1971. Mais il restait encore à réaliser le système de programmation et le système d'exploitation en temps partagé pour au moins 9 terminaux (1 télétype et 8 consoles de visualisation) pour les mini-ordinateurs Mitra 15 (SEMS) et T1600 (TéléMécanique).

Le Mitra 15 de Supélec fut livré au lycée de La Celle Saint-Cloud (premier des 58 lycées équipés) en avril 1973 (voir Figure 1). C'est ainsi que nous avons eu le plaisir de fréquenter Yves Noyelle qui venait y achever la mise au point du système à la faveur (?) des plantages *soft*. Sans compter les plantages *hard*, dus pour l'essentiel à des variations brutales du voltage EDF !

Simple d'usage, de syntaxe française, doté de procédures, de fonctions chaînes, d'une structure de ligne, d'un mode machine de bureau... ce langage, adapté aux mini-ordinateurs Mitra 15 et T1600, fut celui d'une communauté enseignante motivée. Avec LSE, la cohérence de l'expérience des « 58 lycées » devenait possible, les échanges indispensables à la formation des enseignants, à la création de logiciels éducatifs (tout était à inventer) et à la recherche pédagogique, étaient grandement facilités. « *Le système LSE assurait une compatibilité et une portabilité bien supérieures à ce qu'aurait permis alors toute autre combinaison.* » [E. Pélisset, 1985]

Système et langage LSE

Les finalités

Je rappelle que ce qui avait animé les concepteurs du LSD, c'était la volonté de mettre à la disposition des élèves de Supélec un langage simple et convivial sans être rudimentaire. C'est ce qui a convaincu la Délégation à l'informatique du MEN de passer commande pour le LSE.

L'orientation générale retenue par le ministère, à la suite du séminaire de Sèvres (cf. note 4), étant l'apport de l'informatique aux différentes disciplines, il convenait que le langage disposât notamment de fonctions chaînes.

Dans ces perspectives, le LSE a bien joué son rôle. Langage évolué, de syntaxe française, il a prouvé son intérêt pédagogique. Conversationnel et interactif, il favorise la créativité, garantit un apprentissage personnalisé, l'autonomie de l'élève qui peut choisir le rythme de son travail... Voilà ce qui se disait déjà dès les années 70 ! Si les applications numériques ne sont pas négligées, la richesse des fonctions chaînes permet de pleinement satisfaire les besoins de traitement des caractères. Les services

5. J. Baudé, *L'expérience des « 58 lycées »*, bulletin 1024, n° 4, octobre 2014, pp. 105–115, <http://www.societe-informatique-de-france.fr/wp-content/uploads/2014/10/1024-4-baude.pdf>



FIGURE 1. Le Mitra 15 était accompagné de 8 consoles de visualisation permettant un travail en binôme. L'impression se faisait sur un télétype (ASR 33) collectif pourvu d'un lecteur-perforateur de ruban permettant d'entrer et de sauvegarder programmes et données. L'entrée de la « galette » LSE prenait plus d'une heure. Ce n'est pas là mon souvenir le plus agréable. Au lycée de La Celle Saint-Cloud en 1973. On ne voit pas le lecteur de ruban à gauche du télétype. Il a fallu attendre 1977 pour que le lycée soit doté d'un disque souple (*floppy disk*).

fournis sont nombreux et étendus : gestion autonome des fichiers (sans passer par le système d'exploitation), intervention dans le déroulement du programme pour arrêter, modifier, relancer l'exécution, création de procédures internes et externes, etc.

Créé pour des utilisateurs non informaticiens, les élèves et leurs professeurs, il n'a cessé d'évoluer grâce à ces derniers au cours des années. Sa maîtrise par les enseignants en a fait un instrument pédagogique de qualité.

L'EPI, bien consciente de l'intérêt pédagogique du LSE, en fit la promotion tout en ne fermant pas la possibilité d'évolution et d'émergence de nouveaux langages. Ainsi, dans sa participation au schéma directeur (en 1982) on peut lire :

« Le LSE doit être, pour l'instant, conservé. Il est totalement maîtrisé dans son évolution par les enseignants eux-mêmes. Il est apte, par sa normalisation actuelle comme par ses développements en cours (graphique) à répondre aux besoins dans tous les ordres d'enseignement. Il peut être amélioré (Bulletin de l'EPI n° 26, p. 14). Son vocabulaire et sa syntaxe française constituent un argument décisif dès lors que l'introduction de l'informatique est réalisée dans les collèges, les écoles et la formation permanente. Il est urgent de se préoccuper d'installer le LSE sur les micro-ordinateurs français destinés au grand public (l'EPI a pris des contacts avec la Camif et Thomson à ce sujet).

Ainsi la France est en mesure de fournir un modèle complet d'introduction de l'informatique dans l'éducation. Ce modèle peut être un des éléments permettant de retrouver le rôle culturel qu'elle a perdu dans de nombreux pays (des versions du LSE en espagnol et en anglais existent déjà).

Il convient de :

- *Réfléchir dès maintenant à l'évolution ou à la relève du LSE. Créer très vite une commission de réflexion sur les langages.*
- *Réfléchir aussi (et ne pas seulement réfléchir !) aux problèmes que pose la multiplication des micros de toutes marques. Nous devons trouver des solutions (matériel Éducation nationale à bas prix proposant le LSE...). C'est urgent.*
- *Développer des langages-auteurs. Le pluralisme, dans ce domaine, nous semble souhaitable. »⁶*

« Pluralisme », « complémentarité », sont des mots souvent employés par l'EPI qui s'est toujours gardée de tout dogmatisme...

Mais l'association n'était pas sans savoir que tout n'était pas parfait ! L'expérience a eu à souffrir de conditions défavorables liées aux insuffisances du matériel et du logiciel : nombreuses pannes dans certains établissements, lenteur des premières entrées-sorties puis, après l'adoption tardive des disques souples, incompatibilité entre ces supports, incompatibilité entre les systèmes. Sans oublier l'insuffisance des décharges de service (compte tenu du travail à effectuer) et leur arrivée tardive dans les établissements.

Description du langage LSE

Dissipons tout d'abord deux idées fausses assez répandues :

6. Participation EPI au « schéma directeur » (10 juin 1982), <http://www.epi.asso.fr/revue/28/b28p036.htm>, cité dans wikipedia : [https://fr.wikipedia.org/wiki/LSE_\(langage_de_programmation\)](https://fr.wikipedia.org/wiki/LSE_(langage_de_programmation))

(1) Il est erroné de dire, comme on l'entend parfois, que LSE a été inspiré par BASIC. « *En 1970, BASIC (conçu en 1964) était un langage simple mais très rudimentaire, quasiment inexistant en France⁷. En vérité, LSE a deux sources dont la première, injustement oubliée, est le langage PAF (interactif et à syntaxe française), conçu et réalisé par la SEA (Société d'électronique et d'automatisme) pour les ordinateurs CAB 500, qu'elle commercialisait dans les années 60, et la seconde n'est autre que le langage ALGOL, né, lui aussi, vers les années 60.* » [Jacques Hebenstreit, préface de « *Parler LSE* » de Michel Canal.]

(2) Le LSE n'est pas un langage compilé. Il y a traduction d'une ligne de programme entrée en langage source (symbolique LSE) en une chaîne postfixée (en abrégé cpf). Par abus de langage, on appelle « compilation » en LSE le traitement qui réalise l'analyse lexicale, l'analyse syntaxique et la traduction en cpf d'une ligne de programme.

En effet, pour des raisons essentiellement pédagogiques, le LSE a une structure de ligne où chaque ligne est indépendante et analysée (interprétée) dès sa saisie. Le contenu d'une ligne peut être :

- une commande,
- une (ou plusieurs) instruction traitée en mode machine de bureau (elle commence alors par un « ? ») à exécution immédiate,
- une ligne d'un programme LSE (elle commence alors par un numéro) à exécution différée (mode programme).

La ligne ne devra pas comporter plus de 80 caractères y compris le caractère de fin de ligne.

Le système LSE se compose de plusieurs modules : l'initialisateur qui permet d'entrer de plain-pied dans le système (il attend la frappe d'une ligne de programme portant un numéro, une commande ou une requête adressée à la machine de bureau) ; le moniteur ; l'analyseur de commande ; le compilateur et décompilateur ; l'interpréteur (il exécute un programme mis en cpf par le « compilateur ») ; le module machine de bureau (il permet d'exécuter immédiatement une ligne LSE précédée d'un « ? ») ; le module de gestion de fichiers (il stocke en mémoire auxiliaire des fichiers données et des fichiers programmes) ; plus tard, le module graphique.

Les commandes de base sont : BOnjour, LAncer, APpeler, EXécuter, LIster, AUrevoir, etc.

Les instructions du langage LSE (court extrait d'une plaquette de quatre pages correspondant au LSE de 1982⁸) :

- entrées-sorties : LIRE, AFFICHER, CHARGER, GARER...

7. Stéphane Berche, *Comparaison LSE / BASIC*, http://ife.ens-lyon.fr/publications/edition-electronique/revue-francaise-de-pedagogie/INRP_RF056_2.pdf

8. Plaquette LSE, <http://www.epi.asso.fr/blocnote/plaquetteLSE.pdf>

- déclarations : CHAINE, TABLEAU, BOOLEEN, FORME...
- branchements : ALLER EN, SI ALORS SINON, FAIRE...
- interruption : PAUSE, TERMINER...
- exécution : EXECUTER...

Les instructions de déclaration :

- LSE connaît, dans ses dernières versions, quatre types d'objets : les nombres, les chaînes, les booléens et les objets graphiques,
- les fonctions : numériques et chaînes,
- les procédures de deux types : fonctions et sous-programmes.

Mais, compte tenu de la diversité des matériels diffusés au cours des années 70 et 80 (mini-ordinateurs, micro-ordinateurs, nanoréseau...) il convenait, tout particulièrement pour le langage de commande, de se reporter aux manuels des constructeurs. Mais l'objet de cet article n'est pas d'entrer dans les détails du système et du langage. Les collègues intéressés trouveront à la fin de cet article une bibliographie et des articles en ligne.

Voici un extrait du logiciel de simulation (réalisation d'une carte chromosomique en terminale) :

```

APPELER LINKO
1*S LINKOVER J.BAUDE LA CELLE SAINT-CLOUD 04-02-1974
3 AFFICHER [/, ' ETABLISSEMENT D'UNE CARTE FACTORIELLE.' ]
10 CHAINE GS
11 AFFICHER [3/, ' LES 12 GENES LIES SONT : A C H I J K M P
R T U Y' ]
12 AFFICHER [2/, ' TAPEZ 3 GENES (EX : CJR) :'] ; LIRE GS
13 SI LGR(GS)#3 ALORS ALLER EN 14 SINON ALLER EN 16
14 AFFICHER[/, ' SYMBOLE DE GENE INCORRECT.' ] ; ALLER EN 12
16 TABLEAU TOT[6] ; CHAINE SS ; TABLEAU G[5]
18 FAIRE 43 POUR Q3 ← 1 JUSQUA 3
...
49 GARER SS,1, 'LINK' ; GARER TOT,2, 'LINK'
50 EXECUTER 'LL', 51

```

« *Le langage a été révisé en 1983, par Jacques Arzac, pour enfin intégrer la notion de boucle explicite indispensable à toute bonne programmation structurée et se débarrasser de son “ALLER EN” (go to)* »⁹.

9. [http://www.wikiwand.com/fr/LSE_\(langage_de_programmation\)](http://www.wikiwand.com/fr/LSE_(langage_de_programmation)). Voir également LSE83 (Jacques Arzac), http://www.epi.asso.fr/fic_pdf/b38p116.pdf

Le LSE pour quoi faire ?

La création de logiciels pédagogiques par les enseignants

J'ai eu l'occasion d'écrire dans l'article sur « l'opération 58 lycées » (voir note 5) que dans les années 70 tout était à inventer dans le domaine vierge des logiciels pédagogiques. C'était alors le domaine des pionniers en pleine innovation.

Les groupes de recherche INRP dans les différentes disciplines ont, en quelques années, produit en LSE des logiciels originaux dans différents domaines : simulation, traitement de données, exercices d'entraînement, etc. Certains de ces logiciels ont été utilisés ensuite, en version micro-ordinateur, pendant de longues années. À la fin de la décennie 70, une expertise importante avait été accumulée par la communauté enseignante, à la fois sur la conception de logiciels et sur leur mise en œuvre dans les classes¹⁰.

Cette activité s'est poursuivie au cours des années 80. En 1982 a été créée au CNDP une structure consacrée aux logiciels éducatifs : Mission à l'informatique, puis Unité des logiciels éducatifs (ULE).

Cours et clubs d'informatique

Par ailleurs, dans de nombreux établissements, des cours d'informatique étaient organisés (ce fut le cas, dès le début des années 70, au lycée de La Celle Saint-Cloud où une option informatique expérimentale était pilotée par l'INRDP, en langage Fortran)¹¹. Le contenu de ces cours pouvait soit se limiter à la programmation soit être un véritable cours d'informatique avec structure et fonctionnement de l'ordinateur, analyse d'un problème, construction d'un algorithme, programmation...

Des clubs informatiques fonctionnaient également dans de nombreux établissements (lycées, collèges). En témoigne le concours sur le thème « Informatique à l'école » organisé par l'AF CET (Société savante d'informatique, d'automatique et de recherche opérationnelle) avec la participation de l'EPI. Sur l'ensemble des dossiers reçus, 60 programmes étaient écrits en LSE (41 en BASIC, 1 en COBOL, 1 en FORTRAN). Les travaux des 16 lauréats étaient pour 10 d'entre eux écrits en LSE (6 en BASIC). 14 des 16 établissements scolaires qu'ils fréquentaient étaient des lycées équipés de mini-ordinateurs (Mitra 15 ou T1600) ou de micro-ordinateurs¹².

Je ne reviens pas sur l'option informatique des lycées, j'ai abondamment traité du sujet¹³. Je rappelle que les trois langages retenus pour cette option (créée en 1981,

10. Collectif (1981), *Dix ans d'informatique dans l'enseignement secondaire, 1970-1980*, INRP, Recherches pédagogiques, n° 113, 197 pages, http://www.epi.asso.fr/blocnote/Dix_ans_INRP_1981.pdf, et http://lara.inist.fr/bitstream/handle/2332/1250/INRP_RP_81_113op.pdf?sequence=2

11. Étude Pitié-Scherer, http://emnps.net/pdf/1971/etude_pitie-scherer.pdf

12. À propos du concours AF CET « L'informatique à l'école », voir <http://www.epi.asso.fr/revue/22/b22p059.htm>

13. Jacques Baudé, *L'option informatique des lycées dans les années 80 et 90* (en 3 parties chaînées), http://www.epi.asso.fr/revue/histo/h10oi_jb1.htm

supprimée en 1992, rétablie en 1995, et supprimée à nouveau en 1998), et donc pour l'épreuve écrite au baccalauréat, étaient BASIC, LSE et PASCAL.

L'évolution du LSE (mini, micro 8 bits, nanoréseau, 16 bits...)

LSE sur les mini-ordinateurs : Mitra 15 et T1600

L'une des caractéristiques du LSE est qu'il a été créé en vue d'une utilisation conversationnelle, c'est-à-dire que chaque utilisateur peut « dialoguer » avec les différentes ressources du système par l'intermédiaire du clavier du télétype ou de la console de visualisation.

Alternativement, chaque utilisateur dispose de la totalité du système : les travaux de toutes les consoles semblent s'effectuer simultanément et on pourrait croire que la mémoire est divisée en autant de parties qu'il y a d'utilisateurs. Un tel système serait peu performant et laisserait peu de place en mémoire à chacun. Ce n'est pas la mémoire mais le temps que le système consacre aux différentes tâches qui est partagé.

Pour effectuer la gestion de tous les travaux, l'analyse, l'exécution des tâches des utilisateurs, il existe un ensemble de programmes, baptisés « système LSE ». Une partie est en permanence en mémoire (le « résident »). Les autres ressources du système, auxquelles tous les utilisateurs ont recours, sont stockées sur le disque et chargées en mémoire lorsque cela est nécessaire.

Puis apparurent les micro-ordinateurs 8 bits...

Les difficultés économiques entraînent un plan d'austérité. En septembre 1975 la Mission à l'informatique est supprimée. La Direction générale de la programmation et de la coordination (DGPC) supprime les formations et décide, début 1976, l'arrêt des équipements.

L'opération « 58 lycées » continue sur sa lancée sans moyens importants nouveaux donc sans que de nouveaux établissements la rejoignent. Toutefois, le volume des décharges de service et une partie des postes d'enseignants sont conservés. À cela s'ajoute l'évolution rapide des matériels qui change quelque peu la donne. Il faudra attendre février 1979 pour que, à la suite du rapport Nora-Minc, le Président de la République demande au gouvernement d'élaborer un nouveau plan informatique pour accroître l'efficacité et la compétitivité du système économique.

Ce sera, pour le système éducatif, le plan 10 000 micro-ordinateurs (10 000 annoncés...). L'EPI obtient que soit conservé le LSE avec son potentiel de portabilité et que soit sauvegardé le travail important d'une communauté enseignante ayant produit environ 400 logiciels éducatifs.

Pour les matériels, les choix se sont portés sur des micro-ordinateurs 8 bits de constructeurs français aujourd'hui disparus : Logabax (LX500), Société Occitane d'Électronique (X1) – voir Figure 2 –, R2E (Micral). Environ 1 200 de 1979 à 1981. Le LSE, présent dans le cahier des charges des différents constructeurs, sera adapté.



FIGURE 2. Le Système X1, de la Société Occitane d'Électronique

Deux langages sont au cahier des charges : le BASIC et le LSE. L'implémentation de ce dernier par une seule équipe du CNDP assure la compatibilité. Les logiciels écrits en LSE sont transposables d'une machine à une autre de marque différente.

On redécouvre, pour ce qui concerne les logiciels, adaptés non sans peine aux nouveaux matériels, les apports pédagogiques de l'expérience des « 58 lycées » !

Au cours de l'année 1982-83, la bibliothèque diffusée gratuitement par le CNDP est uniquement disponible en LSE. Il s'agit des logiciels provenant de l'opération des « 58 lycées » recueillis par l'INRP.

Un appel d'offre est lancé auprès des enseignants pour une nouvelle production de logiciels éducatifs. Cette procédure, entamée en 1983, s'étalera sur plusieurs années. En plus du LSE, le CNDP soutient le développement d'autres langages comme LISP et LOGO (pour l'école primaire).

Après des épisodes variés et chaotiques, dont le « gel » du plan informatique par Alain Savary, et une opération dans 16 départements (les lecteurs intéressés se reporteront à l'article d'Émilien Péliisset¹⁴), ce sera cette fois le plan 100 000 micro-ordinateurs et 100 000 enseignants à former. Le marché « 4^e tranche » s'ouvre à cinq constructeurs. À la rentrée 1984, plus de la moitié des lycées (650 + 700 LEP) avaient un équipement de l'Éducation nationale.

Puis ce fut le plan « Informatique pour tous », dont j'ai traité dans 1024¹⁵.

Le CNDP est chargé de la production et de la diffusion des logiciels. Pour l'essentiel, il s'agira d'adapter aux PC les 400 logiciels éducatifs développés dans le cadre de l'expérience des « 58 lycées », la grande majorité en LSE. Un interpréteur a été développé à cette fin.

14. Émilien Péliisset, *Pour une histoire de l'informatique dans l'enseignement français, premiers jalons*, <http://www.epi.asso.fr/revue/histo/h85ep.htm>

15. J. Baudé, *Le plan « Informatique pour tous »*, bulletin 1024, n° 5, mars 2015, pp. 95–108, <http://www.societe-informatique-de-france.fr/wp-content/uploads/2015/04/1024-5-baude.pdf>

Les enseignants, seuls ou en équipe, continuent de programmer des logiciels en LSE et à les proposer au CNDP. Après amélioration éventuelle, ce dernier les diffuse gratuitement (seuls les disquettes et le port sont à la charge des établissements). L'esprit du libre fonctionne déjà.

LSE sur le nanoréseau

Je ne reviens pas sur le plan « IPT ». Différentes versions du LSE se succéderont : 2.2, 3.2, 3.3, 3.5, LSE-NR... La portabilité commence à souffrir.

En 1990, on estime à au moins 120 000 le nombre d'ordinateurs Thomson : MO5, MO5NR, MO6... (regroupés en nanoréseaux) et TO7, TO8, TO9... (en postes isolés) dans les établissements scolaires. Ces équipements commencent à être obsolètes et la question de leur remplacement se pose. Le CNDP va s'employer à permettre le transfert des logiciels (dont beaucoup sont en LSE) du format Thomson aux formats PC. Autant dire que les problèmes posés ont été nombreux et redoutables.

Par exemple, dans le catalogue CNDP « Micro-savoirs » (1987), la grande majorité des 160 logiciels en vente fonctionne sous LSE, essentiellement les versions 3.2, 3.5 et LSE84 (ce dernier pour les ordinateurs 16 bits). Dans la rubrique « Langages et outils » est vendue 300 Francs la nouvelle version 3.5 (LSENR) pour MO5, MO6, MO7 et MO7-70 équipés de cartouches RAM et fonctionnant en nanoréseau. Les performances sont améliorées en vitesse d'exécution et étendues au graphique.

LSE sur micro-ordinateurs 16 bits

Le système LSE 8086/8088 a été réalisé par la société Microdur pour les micro-ordinateurs 16 bits. Il fonctionnait sous différents systèmes d'exploitation : CP/M86 pour les SIL'Z 16, MS-DOS pour Persona 1600, Prologue pour Micral 9020, etc. Microdur était une SARL fondée en 1982 par une équipe comprenant de nombreux enseignants membres de l'EPI. Cette équipe, très au fait des besoins du système éducatif, a contribué au développement des réalisations françaises en matière d'informatique pédagogique. L'EPI et Microdur ont publié, en juin 1983, un dossier technique de 230 pages : « *Le système LSE : mieux connaître son fonctionnement sur micro-ordinateur* » :

Le LSE aurait été utilisé par l'École d'application des transmissions de l'armée (j'ai effectivement rencontré il y a quelques années un Mitra 15 à la caserne du Mont Valérien à Suresnes) et un compilateur LSE pour ordinateur CII 10170 et CII IRIS aurait été développé et utilisé dans les universités disposant de ces matériels. Je n'ai pas plus d'informations à ce sujet et suis preneur...

Les publications

De nombreux livres et articles se succèdent au cours des décennies 70 et 80, depuis la brochure sur le LSD (Utilisation du système en temps partagé ; langage L.S.D., École supérieure d'électricité, 58 pages, en 1970), et les publications de



FIGURE 3. « LSE POUR TOUS » (3^e édition)

l'INRP, jusqu'aux ouvrages parus chez plusieurs éditeurs. On se reportera à la bibliographie en fin d'article.

Comme exemple, le sommaire de « LSE POUR TOUS » (1^{re} édition EPI, numéro spécial en supplément au bulletin d'octobre 1981) : Préface et EPIgramme. Pour apprendre à manipuler seul. Ce que l'on traite. Qu'est-ce qu'un programme. Expressions numériques et expressions chaînes. Traitement séquentiel. Itération. Fichiers. Procédures. Langage de commande. De la formulation d'un problème à sa résolution. Annexes (dont quelques conseils en programmation).

En publiant ce travail collectif, l'EPI répond à son objectif : « promouvoir et favoriser toute action de recherche et de formation visant à la diffusion et à l'utilisation de l'informatique dans l'enseignement public ».

Ce numéro spécial sera suivi de plusieurs coéditions CNDP-EPI, dont la dernière paraîtra en juin 1985 (voir Figure 3).

Le déclin et la disparition

Le début des années 80 voit le développement de la micro-informatique. IBM

lance une machine appelée Personal Computer (PC). Microsoft diffuse le système d'exploitation MSDOS. La chute des prix qui caractérise cette nouvelle génération de produits va changer la donne. Des logiciels nouveaux, dits « professionnels », font leur apparition (traitements de texte, tableurs, questionnaires de données...). L'attrait du *made in USA*, souvent doublé d'un *a priori* négatif pour ce qui est français, le fait que le MEN ne se préoccupe guère du suivi d'un langage qu'il a suscité... laissent peu de chance au LSE.

Les mini-ordinateurs – qui ont continué d'être utilisés dans les 58 lycées – furent submergés par la vague des micro-ordinateurs 8 bits sur lesquels furent néanmoins portés le LSE et son environnement.

Pour des tas de raisons, plus ou moins bonnes, le LSE eut des adversaires acharnés.

C'est vrai que dans les années 80, la multiplication des ordinateurs de différentes marques conduit à des versions différentes du LSE qui ont mis à mal ce qui faisait une grande partie de son intérêt, la compatibilité. C'est vrai aussi que la concurrence du BASIC, qui avait évolué et qui était largement présent hors de l'école, pesa lourd dans le désintérêt pour le LSE.

La compatibilité, qui était un des arguments en faveur du LSE, n'existait pas pour les matériels acquis par les établissements sur leurs fonds propres (Apple, Tangy, Commodore, ZX81 de Sinclair...) dont aucun ne disposait du système LSE. Pour les élèves le BASIC était beaucoup plus répandu hors du lycée.

Le désengagement du MEN, déjà évoqué, pour ce langage qu'il avait lui-même commandé, est une illustration, une fois de plus, d'une certaine politique « en dents de scie ».

Ce désengagement est perceptible dès le début des années 80 (opération 10 000 micros). Le rapport Pair-Le Corre¹⁶, octobre 1981, mentionne à peine ce langage (en annexe 3 à propos des stages de formation : « Langage LSE, plus ouverture vers d'autres langages »). Il faudra une intervention appuyée de l'EPI pour que le LSE soit disponible sur le nanoréseau de l'opération « Informatique pour tous ».

On peut aussi regretter que les constructeurs aient fait si peu d'efforts pour s'intéresser à ce langage. Dans le cadre d'une politique d'inspiration libérale lancée en 1986, le secteur privé a pris une place plus importante dans le domaine des logiciels. Le CNDP a vu ses missions dans ce domaine orientées vers la co-production. Celles-ci, assurées dans le cadre d'une « Direction de l'ingénierie éducative », privilégiaient trois axes : outils pour l'enseignant, supports pour auto-formation et l'apprentissage autonome, enrichissement des disciplines.

16. « L'introduction de l'informatique dans l'éducation nationale », rapport de MM. Claude Pair et Yves Le Corre, remis à Alain Savary le 15 octobre 1981, http://www.epi.asso.fr/revue/histo/h81_Pair-Le-Corre.htm

Dans cette phase, le ministère de l'Éducation nationale s'est réservé un rôle d'orientation, par l'intermédiaire de la politique dite de « licences mixtes », inaugurée cette année-là et maintenue ensuite (en fonction des orientations prioritaires, une commission nationale sélectionne des produits, dont les lycées et collèges ne payent qu'une fraction du prix public).

Les effets de cette politique ont été très importants. Elle a rendu disponibles dans les établissements des logiciels professionnels récents. C'est ainsi qu'ont fait leur apparition de nouveaux types de produits logiciels dits « professionnels » qui n'ont plus rien à voir avec le LSE. Corrélativement, elle a conduit à privilégier l'utilisation des « outils » au détriment de l'enseignement d'une discipline informatique.

Pour un musée des systèmes anciens

Je propose ces adresses... J'avoue ne pas les avoir explorées.

— Émulateurs de tous les ordinateurs 8 bits Thomson

<http://dcmoto.free.fr/>

— Documentation sur LSE2000

http://fr.wikibooks.org/wiki/Programmation_LSE

« Cela n'a pas empêché LSE de sombrer rapidement dans l'oubli à la suite de l'abandon de celui-ci par le gouvernement français. Il existe cependant une petite communauté d'utilisateurs et le langage a fait l'objet d'une révision proposée par Luc Goulet (LSE-2000) qui reprend en bonne partie les propositions de 1983 et inclut des améliorations. Notamment, elle ajoute au langage la prise en charge de la programmation orientée objet et aspect. Un wikilivre décrit la proposition LSE-2000 sur wikipedia : [https://fr.wikipedia.org/wiki/LSE_\(langage_de_programmation\)](https://fr.wikipedia.org/wiki/LSE_(langage_de_programmation)) »

— Nasium-lse

<http://nasium-lse.ogamita.com>

« Dans un but historique et pseudo-muséologique, nous présentons une implémentation du système LSE, NASIUM LSE, qui devrait permettre de faire tourner les vieux programmes LSE, et d'avoir une impression de ce qu'était l'informatique dans les années 1970. »

— Wikiwand

[http://www.wikiwand.com/fr/LSE_\(langage_de_programmation\)](http://www.wikiwand.com/fr/LSE_(langage_de_programmation))

Conclusion

Le langage LSE avait des qualités incontestables, et il a évolué au cours des années, mais il s'est heurté à un ensemble de contextes défavorables comme par exemple :

- le manque de constance et de cohérence des politiques informatiques et les décisions prises par quelques-uns, au sein du MEN, sans évaluations ni concertations,
- la concurrence directe du BASIC, tellement mieux puisqu'il vient d'ailleurs, comprenez d'outre-atlantique,
- l'arrivée des logiciels « professionnels » et la priorité donnée par le ministère à l'utilisation des « outils » au détriment de la programmation dans le cadre d'un enseignement de l'informatique...

Mais on va m'accuser d'être partial, aussi je laisse le dernier mot à *Science et Vie Micro* de mai 1986 :

Plaidoyer pour un langage méprisé (extrait) : « *Ce langage propre à l'Éducation nationale française, beaucoup décrié, continue malgré tout son chemin. On se rend compte aujourd'hui que ceux qui avaient vaillamment défendu le LSE n'avaient peut-être pas tout à fait tort.* »

Suit toute une série d'arguments en faveur de ce langage : syntaxe, française, structure de ligne, procédures, chaînes, graphismes « étonnants ».

Et le coup de pied de l'âne : « *La méfiance des informaticiens qui ont souvent critiqué le LSE sans en connaître vraiment les possibilités* »...

S'il n'y avait eu que les informaticiens !

Mais les choses sont en train de changer... et l'on redécouvre la programmation dans les enseignements scolaires.

Bibliographie

Utilisation du système en temps partagé (langage L.S.D.), École supérieure d'électricité, polycopié n° 2162 et brochure n° 2193, 58 pages, 1970.

Un Langage Symbolique destiné à l'Enseignement : le LSE, par J. Hebenstreit et Y. Noyelle, Congrès AFCET, Grenoble, novembre 1972.

LSE manuel d'utilisation, numéro spécial du bulletin de liaison « l'informatique et l'enseignement secondaire » de l'INRDP, Section Informatique et enseignement (SIE), 108 pages, avril 1973.

Un langage symbolique destiné à l'enseignement : LSE, par J. Hebenstreit et Y. Noyelle, bulletin n° 6 de l'EPI (1973), <http://epi.asso.fr/revue/06/b06p010.htm>

Langage symbolique d'enseignement, manuel d'utilisation, numéro spécial du bulletin de liaison « l'informatique et l'enseignement secondaire » de l'INRDP, Section Informatique et enseignement (SIE), janvier 1975, 202 pages. Le LSE Pour la version 4 (T1600) et la version 8 (Mitra 15), rédaction terminée en décembre 1974.

LST : A Highly Adaptative Real-Time Time-Sharing System, par S. Berche, J. Hebenstreit et Y. Noyelle, Symposium IFAC/IFIP, Tallinn, Mai 1976.

Le LSE, son histoire, son développement, par Y. Noyelle, Éducation 2000 n° 16, Institut Supérieur de Pédagogie, Paris, 1980.

LSE POUR TOUS, numéro spécial supplément au bulletin de l'EPI, collectif, 210 pages, 2 éditions successives, octobre 1981.

LSE POUR TOUS, coédition CNDP-EPI (1982), même contenu plus mise en route du Micral (R2E) et du Logabax (LX 529).

LSE POUR TOUS, deuxième coédition CNDP-EPI (1983), même contenu plus deux chapitres : graphique et compléments, de juin 1983.

Norme expérimentale Z65-020. LSE : Langage Symbolique d'Enseignement, AFNOR, Paris, décembre 1983.

Le système LSE, mieux connaître son fonctionnement sur micro-ordinateur, dossier EPI n° 3, EPI + Microdur, 230 pages, juin 1983.

Le LSE 8086/8088 (pour débiter avec le SIL'Z 16 et le PERSONA 1600, manuel utilisateur, manuel de référence, d'après la norme Afnor Z65-020, classeur EPI-Microdur, 220 pages, septembre 1984.

LSE POUR TOUS, troisième édition CNDP-EPI, plus deux chapitres : LSE sur TO7 - MO5, et LSE sur machines 16 bits, 1985.

String Handling Facilities of the LSE language, par Y. Noyelle, SIGPLAN, août 1984,

LSE 83, par J. Arsac, bulletin de l'EPI n° 38, pp. 116–139, juin 1985 (sur le site EPI).

Programmation en LSE, par Yves Noyelle et Stéphane Berche, Éd. PSI, 1979.

Lire LSE, par C. Lafond et P. Muller, Éd. Nathan.

Parler LSE, par Michel Canal, Éd. Eyrolles, 1982.

LSE : Programmer en français, par A. Thomazo, Éd. Scodel, 1981.

Exercices d'application du LSE, par A. Billès Éd. Eyrolles.

L'abc du LSE, par C. Cohort Éd. Eyrolles.

Le LSE pour tous, TO7 et MO5, Éd. Magnard.

[Certains de ces ouvrages parus chez les éditeurs sont disponibles d'occasion sur des sites de vente en ligne.]

14 documents à Bibliothèque nationale de France (BNF) :

http://data.bnf.fr/11940440/lse__langage_de_programmation/

<http://data.bnf.fr/documents-by-rdt/11940440/a/page1>

Langage LSE sous siris 7 / siris 8 – manuel d'utilisation, par P. Fischer et P. Mercier (i.u.c.a) référence : notc0031, mai 1978, Institut universitaire de calcul automatique de Lorraine.

Langage symbolique d'enseignement LSE 15 – manuel d'utilisation, École d'application des transmissions, Direction de l'instruction, diffusion interne à des fins pédagogiques seulement.

Sans oublier d'aller consulter la : « *List of French inventions and discoveries* », chapitre 9, dans lequel le LSE figure avec Champollion, Braille, Micral et le Minitel !
https://en.wikipedia.org/wiki/List_of_French_inventions_and_discoveries#Communication_.26_Computers

Articles et documents sur le site de l'EPI (<http://www.epi.asso.fr>) *accessibles avec votre moteur habituel :*

site:epi.asso.fr LSE

site:epi.asso.fr langage symbolique d'enseignement



Lettre ouverte à Mme Najat Vallaud-Belkacem

Maurice Nivat

Le président du Conseil supérieur des programmes a remis vendredi 18 septembre 2015 le projet de programmes pour le primaire et le collège (cycles d'enseignement 2, 3 et 4) à la Ministre de l'Éducation nationale, Madame Najat Vallaud-Belkacem. L'utilisation d'outils numériques y figure dès le cycle 1 et l'informatique est présente au cycle 4 en mathématiques (algorithmique et programmation) et en technologie (programmation, réseaux et protocoles).

Le 5 octobre dernier, la SIF et l'EPI publiaient un communiqué commun : Enseigner l'informatique : il est temps de s'inscrire dans la durée¹, se réjouissant de cette introduction et alertant la Ministre sur l'importance de la formation des enseignants amenés à intervenir dans ces disciplines.

À travers cette lettre ouverte à Madame Najat Vallaud-Belkacem, Maurice Nivat revient sur l'importance majeure de cette question.

Madame la Ministre,

Vous avez souhaité l'introduction du numérique à tous les niveaux de l'enseignement du primaire à la terminale et voilà qui est fait, au moins dans les programmes que le CSP a rendus publics. C'est une bonne chose. La SIF et l'EPI, qui ont toujours défendu la nécessité de donner à tous les jeunes français les connaissances et les compétences voulues pour bien vivre dans le monde numérique qui est devenu le nôtre, ne peuvent que se réjouir de cette introduction.

1. <http://www.societe-informatique-de-france.fr/2015/10/enseigner-linformatique-il-est-temps-de-sinscrire-dans-la-duree/>

Ils se réjouissent particulièrement du fait que l'informatique, qui est la partie scientifique et technique de l'édifice numérique, trouve la place qui lui est due dans ces programmes, conformément aux recommandations contenues dans le rapport de l'Académie des sciences émis en 2012 : les grandes applications numériques qui intéressent la société toute entière reposent en effet sur le développement de systèmes contenant une large part d'informatique sous sa forme la plus pointue, rendu possible seulement par les résultats les plus récents de la recherche en informatique et par un nombre suffisant de développeurs de grande qualité.

Nous pensons que le but de l'enseignement du numérique dans les écoles, les collèges et les lycées n'est pas de transformer tous les jeunes en informaticiens mais de leur donner conscience de l'enjeu formidable que constitue le numérique et des transformations qu'il induit dans la société, comparables à ce que furent, en leur temps, la machine à vapeur et le chemin de fer, la conquête de l'air, l'enregistrement et la restitution des sons, le téléphone et la TSF. Si certains garçons et filles trouveront là une vocation (c'est souvent à l'adolescence, entre la quatrième et la première, que les vocations se forment), tous ont besoin de savoir que l'informatique existe et n'avance pas toute seule, qu'il y a des raisons et des gens qui la font avancer, ainsi que beaucoup de travail déjà effectué, et sans doute encore plus à venir.

Les programmes du CSP nous paraissent bien adaptés pour atteindre ce but à une condition, évidemment primordiale, voire *sine qua non* : il faut qu'ils soient enseignés par des enseignants ayant eux-mêmes une connaissance et une vision de l'informatique qui ne soit pas trop fragmentaire ou anecdotique.

La SIF et l'EPI s'indignent donc qu'une fois de plus le dispositif destiné à faire rentrer le numérique et l'informatique dans la panoplie des matières enseignées ne s'accompagne pas de la création de véritables professeurs d'informatique, agrégés ou capésiens, longuement formés et dont l'informatique serait la matière principale, c'est-à-dire celle sur laquelle ils seront évalués, jugés, et éventuellement promus.

De tels enseignants, professeurs et maîtres de conférence, existent dans l'enseignement supérieur. Certains, qui ont commencé il y a longtemps, quand l'informatique balbutiait encore, déjà à la retraite, peuvent témoigner qu'enseigner l'informatique, qui ne cesse de se développer, pendant quarante ans suffit largement à occuper une vie d'homme ou de femme. C'est une erreur de croire que parce qu'ils enseigneraient à plus bas niveau, à des élèves plus jeunes, des choses forcément plus simples, les enseignants chargés de l'informatique au collège et au lycée pourraient être formés en peu de temps : le plus souvent il ne feraient que régurgiter des connaissances insuffisamment assimilées et il leur manquerait la vision globale (de plus en plus difficile à acquérir tant l'informatique étend son champ et son domaine) permettant de donner du sens aux fragments assez disparates qui seront présentés aux élèves. Et il leur serait bien difficile de communiquer aux élèves un enthousiasme pour la discipline informatique que, faute de la maîtriser convenablement, ils n'éprouvent pas eux-mêmes.

Certes, pour commencer dès cette rentrée scolaire, il faut faire avec ce que l'on a, et sans doute faudra-t-il du temps avant que tous nos élèves soient mis en présence de véritables professeurs d'informatique, mais nous vous conjurons, Madame la Ministre, de déclencher le mouvement vers la création de corps de professeurs d'informatique formés comme le sont ceux des autres disciplines enseignées, mathématiques, physique, ou biologie : si on ne commence pas maintenant on renvoie l'enseignement nécessaire aux calendes grecques car (comme l'expérience ISN nous l'a appris) le vivier des professeurs d'autres disciplines, assez altruistes pour se charger de l'enseignement d'une discipline qu'ils connaissent mal en plus de leur discipline principale, est déjà en train de se tarir.

L'ampleur même des enseignements proposés par le CSP interdit de penser qu'ils pourront entièrement reposer sur le bénévolat des enseignants d'autres disciplines.

Vous avez fait un pas important, Madame la Ministre, faites l'autre, tout aussi nécessaire : ce ne sont pas les informaticiens qui vous en seront reconnaissants, c'est toute la France.

MAURICE NIVAT

Professeur Honoraire à l'université Paris 7 Denis Diderot,
membre d'honneur de la Société informatique de France,
membre d'honneur de l'association Enseignement Public et Informatique.



Enseigner l'informatique à des linguistes (filières TAL)

Isabelle Tellier¹

Cet article est basé sur l'exposé donné par Isabelle Tellier, Professeure en Sciences du langage et précédemment Professeure en Informatique, lors des journées pédagogiques de la SIF qui se sont déroulées à Paris (CNAM) du 23 au 24 juin 2015. Le thème de ces journées était « L'enseignement de l'informatique pour les humanités et les sciences sociales ».

Le traitement automatique des langues (TAL) est une discipline éminemment pluridisciplinaire, qui requiert une vraie double compétence en informatique et en linguistique. Des filières de niveau Master dédiées à ce domaine existent dans des départements de sciences du langage, où elles recrutent essentiellement des linguistes (c'est le cas à Sorbonne Nouvelle – Paris 3 où j'enseigne). Des initiations sont également souvent proposées au niveau licence (L2 ou L3). Comment se passent les enseignements d'informatique dans ce contexte ? Quelle informatique privilégier, comment y intéresser les étudiants ?

La chance de l'informatique, c'est que son histoire est très liée à celle de la linguistique : les deux disciplines ont beaucoup de points communs et de références communes (Chomsky en est une illustration exemplaire). Leurs évolutions récentes ont aussi suivi des itinéraires parallèles. En linguistique de manière générale, et en TAL en particulier, comme dans la plupart des branches de l'intelligence artificielle,

1. Université Sorbonne Nouvelle – Paris 3.

on est passé ces 30 dernières années de descriptions fondées sur des modèles formels/symboliques à base de règles (analyses syntaxiques avec des grammaires formelles, analyses sémantiques avec des représentations logiques...) à des études empiriques fondées sur l'usage (linguistique de corpus, statistiques textuelles, fouille de données réelles...). La linguistique est une discipline qui revendique un ancrage scientifique fort. Avec ma formation initiale en informatique, je n'ai pas eu de difficultés à comprendre les problématiques et les méthodologies de recherche de mes collègues linguistes. Et inversement, convaincre les linguistes qu'ils ont tout à gagner à acquérir de bonnes compétences en informatique est aussi relativement facile, au moins pour les linguistes professionnels. Mais les étudiants ont souvent des réticences vis-à-vis des mathématiques (ils se sentent plus « littéraires » que scientifiques), il faut en tenir compte.

Les initiations en licence sont généralement orientées « données » et « outils » : codage des textes, formats de fichiers, initiation à la statistique textuelle, requêtage par expressions régulières, analyses linguistiques en ligne, utilisation d'étiqueteurs ou d'analyseurs syntaxiques... De nombreuses notions fondamentales de l'informatique peuvent déjà être illustrées par ce type d'enseignements. La programmation est abordée après coup, une fois constatées les limites des outils. Ces cours sont spécifiques. De fait, l'enseignement de la programmation que j'ai reçu ne serait pas du tout adapté à ce contexte. J'ai appris à programmer par les nombres, en implémentant des traitements qui illustraient les cours de mathématiques : calcul des termes d'une suite (Fibonacci et compagnie), triangle de Pascal, tri de nombres, récursivité de la factorielle, résolutions d'équations (polynômes, pivot de Gauss, dichotomie, méthode de Newton...), manipulation de vecteurs ou de matrices... autant de problèmes qui intéresseraient bien peu mes étudiants. L'enseignement de l'informatique en France est clairement l'héritier des mathématiques. Avec les linguistes, il faut s'appuyer sur un autre ancrage, quitte à faire passer certaines notions mathématiques en douce.

Un premier terrain traditionnel, plus « neutre » en termes disciplinaires, est fourni par le domaine des jeux. Les jeux sont des sources d'exercices et de projets quasi infinis, toujours motivants pour les étudiants. Ceux qui utilisent la langue sont nombreux : jeu du pendu, mots croisés/fléchés/mélangés, codes secrets, Scrabble (le comptage de la valeur d'un mot posé sur des cases « lettre compte double/triple » équivalait d'ailleurs à réaliser un produit scalaire entre deux vecteurs, mais on n'est pas obligé de le savoir pour le programmer), jeu des « chiffres et des lettres », MasterMind avec des lettres (où il faut découvrir un mot caché), jeux oulipiens... Seul imprévu rencontré lors d'examens : ces jeux à base de lettres sont inconnus des étudiants qui ont appris à écrire avec des caractères idéographiques, il ne faut pas négliger l'explication des règles...

De manière générale, pour enseigner la programmation aux linguistes, il est judicieux de considérer que la donnée fondamentale est moins le nombre que la chaîne

de caractères. J'évoque ci-dessous quelques types d'exercices permettant d'introduire certains concepts fondamentaux de la programmation :

- Conditions : exercices de conjugaisons ou de déclinaisons (pour des langues qui s'y prêtent) automatiques, et autres flexions (programmer la règle de mise au pluriel des noms communs en français, en tenant compte des différents cas particuliers et exceptions, par exemple), jeux sur les codes ASCII pour transformer des majuscules en minuscules et inversement...
- Boucles : jeu sur le codage/décodage de chaînes avec le codage de type « César » (décalage des lettres de l'alphabet, ce qui permet au passage d'introduire les modulo quand on dépasse 'z'), comptage de certains caractères/chaînes dans d'autres chaînes...
- Tableaux, structures de données complexes : compte d'occurrences de lettres/mots, stockage de dictionnaires monolingues/bilingues ou de méta-données d'un livre. Par ailleurs, les structures d'arbres abstraits (syntaxiques) sont familières aux étudiants de sciences du langage, les manipuler leur est donc naturel.
- Algorithmes de tri, introduction à la complexité : le tri des livres d'une bibliothèque sera plus parlant pour ces étudiants que le tri d'un ensemble de nombres.
- Récursivité : au-delà des traditionnels palindromes, la plupart des programmes de manipulation de chaînes de caractères peuvent aussi s'écrire de manière récursive. Programmer des grammaires formelles peut bien sûr les séduire (mais, en dehors des langages réguliers, elles sont beaucoup plus faciles à utiliser en mode « génération aléatoire » qu'en analyse). Elles offrent l'occasion de faire le lien entre la récursivité de la langue (par exemple le fait que des groupes nominaux peuvent inclure d'autres groupes nominaux) et celle des programmes qui les modélisent. Ainsi, pour générer les phrases incluant toutes sortes de « liens familiaux » décrites par l'automate de la Figure 1, le mieux est d'écrire autant de fonctions que d'états dans l'automate, et de les faire s'appeler mutuellement.

Pour des projets plus complexes et complets, on peut par exemple proposer la programmation d'un correcteur orthographique à partir d'un dictionnaire de formes fléchies et du calcul de la distance d'édition entre chaînes de caractères, ou celle d'un petit moteur de recherche, en partant du comptage des mots dans un texte et du calcul de la « distance » entre deux textes à partir de leurs mots communs (distances de Jaccard, de Dice, proximité cosinus... tiens, les maths reviennent encore par la fenêtre). En master, dans ces formations, les cours de fouille de textes que je donne sont les mêmes que ceux que j'avais préparés pour les filières informatique.

Les goûts et dispositions des étudiants de sciences du langage pour ces enseignements sont évidemment variables. Mais leurs études les ont bien préparés, en les habituant à traiter les textes comme des séquences structurées de symboles. Ils ont

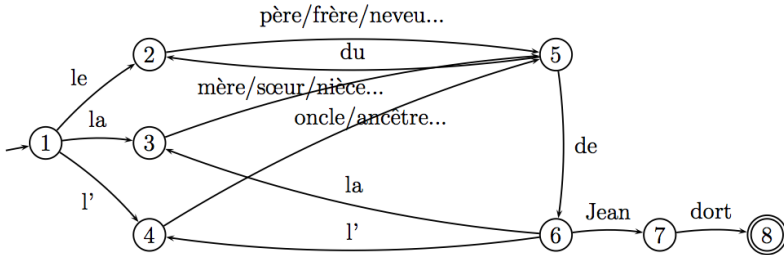


FIGURE 1. Automate décrivant un ensemble de phrases.

appris à les manipuler et à les analyser indépendamment de leur sens, c'est un point de départ précieux pour assimiler les bases de l'informatique. J'ai souvent constaté qu'ils sont sensibles, plus que les étudiants informaticiens, aux limites des outils de fouille de textes et d'ingénierie linguistique actuellement disponibles et savent mieux les critiquer. Ils rédigent aussi généralement leurs projets bien mieux qu'eux... Notons au passage que les ratios filles/garçons dans ces filières sont presque inverses de ceux observés en informatique (90 % vs. 10 %, environ...). À défaut de recruter des filles pour faire de l'informatique, je me réjouis au moins d'enseigner l'informatique aux filles là où elles sont, et de leur montrer qu'elles peuvent aimer ça ! Je peux ainsi attester que, chaque année, d'excellent(e)s étudiant(e)s se révèlent. Je connais plusieurs linguistes ayant effectué une thèse de TAL qui ont obtenu, in fine, leur qualification en section 27. Mais, en dehors de ces cas particuliers, il n'est pas douteux que l'informatique va prendre une place de plus en plus importante dans la formation des linguistes. L'expérience que je retranscris ici devrait aussi, à mon sens, encourager l'enseignement de la programmation à destination des littéraires : à condition de partir de leurs centres d'intérêts, ils peuvent se passionner pour le sujet.



Rapport d'activité de la formation ISN dans l'académie de Montpellier de septembre 2011 à juin 2015

Olivier Cogis¹, Jérôme Palaysi² et Richard Terrat³

Introduction

L'enseignement de l'informatique a fait son apparition dans les lycées par la petite porte : aux trois *spécialités* offertes jusqu'alors⁴ aux terminales S des lycées d'enseignement général, s'est ajoutée en septembre 2012 la spécialité *Informatique et Sciences du Numérique* (ISN).

Les autorités de l'Éducation nationale ont décidé de confier l'encadrement des groupes d'élèves ayant choisi la spécialité ISN à des enseignants volontaires issus des « [...] disciplines scientifiques (mathématiques, sciences physiques et chimiques) et technologiques (sciences et techniques industrielles, etc.)⁵ » préalablement *habilités* à l'issue d'une formation pédagogique, chaque académie étant appelée à en définir les contenus et modalités et chargée d'en assurer la mise en place et le suivi.

Dans l'académie de Montpellier, la formation ISN à destination des enseignants des lycées a été mise en place à la rentrée 2011 sous la double responsabilité consensuellement arrêtée :

- responsabilité académique d'une IA-IPR de Mathématiques, chargée de mission par le recteur,

1. Professeur honoraire des Universités en Informatique à l'Université de Montpellier.

2. Maître de Conférences en Informatique à l'Université de Montpellier.

3. Maître de Conférences honoraire en Informatique à l'Université de Montpellier.

4. *Mathématiques* (Maths), *Physique-Chimie* (PC) et *Sciences de la Vie et de la Terre* (SVT).

5. http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57998

— responsabilité pédagogique d'une équipe universitaire, constituée de trois enseignants-chercheurs en Informatique de l'Université de Montpellier 2 (dont un en activité et deux à titre honoraire)⁶.

Cette formation a pris fin, du moins sous cette forme et sous la responsabilité de cette équipe, au bout de quatre ans d'exercice.

Les formations ISN à l'échelle nationale ont vécu indépendamment les unes des autres, et les autorités de l'Éducation nationale n'ont pas, à notre connaissance, cherché à en faire le bilan ni à en unifier les contenus et pratiques.

En rendant compte de l'expérience qui s'est déroulée dans l'académie de Montpellier, nous souhaitons alimenter, à notre mesure, la réflexion et la discussion sur la *formation des maîtres* à l'informatique.

Ce document évoque successivement les points suivants :

- les modalités de cette formation,
- ses contenus (détaillés en annexe),
- quelques éléments de réflexion suscités par cette expérience.

Moyens pédagogiques et mise en œuvre

Les moyens arrêtés par les autorités académiques

Les principales dispositions décidées par les autorités académiques de Montpellier pour la mise en place de la formation des enseignants à l'ISN ont été les suivantes :

- une formation répartie en deux modules successifs, ISN1 et ISN2 :
 - (1) le premier module menant à l'*habilitation*,
 - (2) le second module permettant aux habilités d'augmenter leurs connaissances (en largeur comme en profondeur) ;
- un enseignement à distance basé sur une plate-forme d'apprentissage en ligne gérée par l'académie (plate-forme Moodle) ;
- des sessions, ISN1 comme ISN2, dotées chacune de 3 journées en présentiel⁷.

En revanche, les contenus de la formation et leur mise en œuvre ont été laissés à la responsabilité de l'équipe pédagogique.

À noter qu'aucune décharge de cours n'a été prévue pour les enseignants participant à la formation.

6. Les trois signataires de ce rapport.

7. À l'exception de la dernière session ISN1 dotée de 5 jours, une évolution sur laquelle nous reviendrons.

La mise en œuvre

De septembre 2011 à juin 2015, se seront déroulées :

- 5 sessions de formation ISN de niveau 1,
- 2 sessions de formation ISN de niveau 2.

Le nombre de mois en distanciel a été variable selon les sessions.

Les journées en présentiel ont été dévolues à :

- la présentation des thèmes informatiques à venir dans les semaines suivantes,
- un retour sur les thèmes étudiés précédemment, lorsque les échanges sur les forums y incitaient,
- selon les circonstances, une discussion ouverte avec les stagiaires sur des sujets variés.

La plate-forme Moodle a été utilisée lors des différentes sessions pour :

- mettre à disposition des ressources (notes de cours et diaporamas),
- proposer des activités (tests, visant à vérifier l'appropriation des concepts de base, et forums d'exercices et de discussions).

Tous les documents mis à disposition sur la plate-forme d'apprentissage en ligne sont originaux⁸.

Au total des 6 sections⁹ des sessions de niveau 1 et des 8 sections des sessions de niveau 2, ont été mis à disposition sur la plate-forme d'apprentissage en ligne :

- 7 tests,
- 31 diaporamas,
- 9 notes de cours ou compléments.

Parallèlement, ont été proposés et animés sur cette plate-forme :

- 15 forums d'exercices,
- 15 forums de discussions thématiques,

ayant donné lieu à quelque 2000 interventions des membres de l'équipe pédagogique.

L'observation de l'activité des différentes sessions¹⁰ nous permet d'affirmer que plus de 180 professeurs ont participé aux modules de niveau 1, et plus de 80 aux modules de niveau 2.

8. À l'exception de quelques documents et d'un test conçus par une équipe de l'université de Limoges et mis en ligne pour permettre le démarrage de la première session d'ISN1 dès septembre 2011.

9. En *Moodle*, les « sections » permettent de subdiviser un « cours ».

10. Nous ne disposons pas de statistiques administratives. Nous croyons cependant savoir qu'au moins 125 professeurs ont été habilités.

Contenus de la formation

Les contenus de cette formation ISN ont été discutés puis arrêtés par l'équipe pédagogique en accord avec la chargée de mission académique et conformément au programme officiel *Enseignement de spécialité d'informatique et sciences du numérique de la série scientifique – classe terminale*¹¹ et dans l'esprit du rapport de l'Académie des sciences *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*¹² dont nous partageons l'esprit et le contenu et dont nous extrayons les quelques lignes (c'est nous qui soulignons) :

« Même si l'informatique est un domaine où la création est rapide et où les produits deviennent rapidement obsolètes, cette création et ces produits s'appuient sur des connaissances fondamentales et des savoir-faire stables. En particulier, la science informatique repose sur de grands concepts unificateurs. Ainsi, le récent programme de la spécialité *Informatique et sciences du numérique*¹³, en terminale scientifique, s'articule autour de quatre concepts et de leurs interactions : *algorithme, langage, information et machine*. »

Nous détaillons en annexe les titres des sections constituantes des modules de formation à l'ISN réparties en deux niveaux, en exprimant succinctement l'esprit.

Réflexions

La formation ISN qui a eu lieu dans l'académie de Montpellier au cours des quatre années scolaires 2011-2015 a revêtu le double aspect d'être à la fois ambitieuse quant à son contenu et exigeante pour les participants¹⁴.

L'équipe pédagogique encadrante s'est fixé comme objectif une transmission des connaissances et des savoir-faire constituant un socle suffisamment solide pour permettre aux participants de pouvoir progresser à titre personnel tout en encadrant des groupes d'élèves de spécialité ISN, et suffisamment large pour pouvoir correctement situer les questions de ces mêmes élèves auxquelles ils n'auraient pas de réponse suffisamment précise à leur fournir dans l'instant.

Il en est résulté que l'acquisition de ces connaissances et de ces savoir-faire a exigé de la part des participants des efforts considérables sachant qu'aucune décharge ne leur a été accordée pour cette formation. Néanmoins les forums ont enregistré un niveau de participation d'une ampleur parfois surprenante et des progressions certaines ont pu y être constatées, bien au-delà du simple *satisfecit*.

11. Bulletin officiel spécial n° 8 du 13 octobre 2011, http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57572

12. http://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf.

13. Opus cité.

14. De fait, on peut considérer que le programme de la formation proposé est d'un niveau proche de celui d'une licence d'informatique.

Sur les contenus

L'origine des participants était assez hétérogène, et leurs points forts comme leurs difficultés se sont donc différemment répartis. Nous en évoquons quelques points saillants, en gardant à l'esprit que les apprenants volontaires sont des professeurs en activité.

L'appropriation pas toujours immédiate des booléens. Une illustration y suffira : le passage de

$$\mathbf{si} A[i] = 0 \mathbf{ alors } \text{trouvé} \leftarrow \mathbf{vrai} \mathbf{ sinon } \text{trouvé} \leftarrow \mathbf{faux}$$

à

$$\text{trouvé} \leftarrow A[i] = 0$$

n'a pas toujours reçu l'accueil enthousiaste qu'il nous semble pourtant mériter.

La spécification des fonctions et des procédures. Chacun connaît la difficulté à obtenir des étudiants :

- (1) qu'ils spécifient leurs fonctions et procédures,
- (2) qu'ils les spécifient « correctement », et alors « suffisamment »¹⁵.

Nous avons fortement insisté sur cette nécessité, en mettant de plus en avant que si de nombreuses difficultés attendaient là les élèves de terminale, l'exercice participait fortement à la constitution de leurs facultés d'abstraction, dans une phase cruciale de leur développement intellectuel et scientifique.

Force nous a été de constater que, au-delà du bien connu « oui, oui, je les ferai après, là je programme » auquel tous n'échappaient pas, les professeurs eux-mêmes ont parfois, et même souvent, éprouvé de grosses difficultés à se soumettre à l'exercice¹⁶.

15. À savoir :

- (a) « correctement » : que le sous-programme satisfasse ses spécifications pour tous les paramètres effectifs qui les respectent,
- (b) « suffisamment » : par exemple, « retourner vrai » est le corps de fonction le plus économique permettant de satisfaire la spécification « Résultat : retourne vrai si le tableau A contient au moins un 0 ».

16. Difficultés qui peuvent commencer par celle d'admettre qu'un paramètre formel auquel les spécifications ne font pas référence n'a pas lieu d'exister.

Notons que nous avons constamment affiché notre refus de lire les sous-programmes dépourvus de spécifications.

Les fonctions métier vs les interfaces et les interfaces en elles-mêmes. Il a fallu parfois insister pour obtenir que les instructions d'E/S soient bannies des fonctions et procédures non expressément dédiées à l'interfaçage¹⁷.

Ensuite, la question a été de mettre en évidence qu'il était vain, aussi bien pour les participants que pour leurs futurs élèves, de chercher à sophistiquer les interfaces au-delà d'un certain stade (tentatives destinées au traitement de « toutes » les erreurs de saisie possibles, recherche de « beaux » affichages...).

De fait, les projets ont été l'occasion pour certains de concevoir des interfaces graphiques, malgré l'investissement technique nécessaire.

On peut le comprendre, car certaines réalisations ont dû fortement satisfaire leurs auteurs par leur indéniable qualité.

Mais nous n'avons pas caché qu'à notre avis relevait d'une erreur stratégique le choix d'enseigner ces techniques à des élèves au détriment d'un travail sur la décomposition d'un programme en sous-programmes résultant de la mise en évidence de différents niveaux d'abstraction, sur une spécification minutieuse de ces sous-programmes, ou encore sur un choix judicieux¹⁸ de commentaires, sans parler de justification d'algorithmes sous-jacents.

L'architecture de von Neumann et la programmation en langage machine. Une fois exposé le schéma de l'architecture et donné un jeu d'instructions machine réduit, en particulier sans adressage indirect ni indexé, la trace d'un petit programme effectuant la somme d'un tableau d'entiers dont les valeurs sont rangées dans des mots mémoire successifs a sans surprise suscité l'étonnement. Qu'un mot mémoire puisse être tantôt décodé comme une instruction, tantôt utilisé comme opérande pour l'unité arithmétique et logique et, au final, qu'il ne soit donc pas plus une donnée qu'une instruction, et réciproquement...

Au vu des discussions apparues, nous persistons à trouver l'exercice salutaire.

D'une façon générale. Les quelques points brièvement évoqués ci-dessus ne devraient pas, à vrai dire, surprendre grand monde. Nous les avons présentés dans l'optique de contribuer aux réflexions et discussions sur comptes-rendus des expériences de ce type que la période vient de vivre.

17. L'argument comme quoi il n'est pas du ressort d'une fonction de maîtriser l'usage qui peut être fait du résultat de ses calculs – par exemple, et pour la même fonction, le résultat peut une fois servir au calcul effectué par une autre fonction, une autre fois être inscrit dans une base de données distante – a le plus souvent fini par l'emporter auprès des participants.

Il est vrai que certains d'entre eux ont pu être fâcheusement influencés par la regrettable décomposition d'un « algorithme simple » en ses trois « éléments de base » que seraient « Les trois étapes », à savoir « La préparation du traitement », « Le traitement » et « La sortie des résultats », décomposition malheureusement proposée en sa page 6 par le document *Éduscol [DGESCO], Mathématiques, Lycée, Ressources pour la classe de seconde – Algorithmique*. Comprendre : la saisie des données et l'affichage de ses résultats font partie intégrante d'un algorithme. No comment.

18. Par opposition, le commentaire « $i \leftarrow i + 1$ # on incrémente i » n'est pas jugé judicieux.

Sur la gouvernance et son impact

Un rapide historique. Du fait d'un changement de responsables académiques de la formation ISN, celle-ci a connu une évolution qu'on peut schématiser en trois phases :

- (1) deux années de mise en place et de stabilisation (2011/2012 et 2012/2013),
- (2) une année de maintien des modules de formation mais de modification des jurys d'habilitation (2013/2014),
- (3) une année de modification des modules de formation (2014/2015).

Le recteur de l'académie de Montpellier en place en 2011 s'est révélé être un partisan déterminé de l'introduction de l'informatique dans les lycées, fût-elle alors limitée à la création de la spécialité ISN pour les terminales S. Il a chargé de mission une IA-IPR de Mathématiques avec la triple responsabilité d'inciter les proviseurs de lycée à ouvrir des groupes ISN, de faire le recensement des professeurs volontaires à l'encadrement de ces groupes et d'organiser la formation de ces professeurs. Cette responsable académique, volontaire pour assumer cette charge, adhère à l'idée que l'informatique est une discipline scientifique et que les experts se trouvent dans les universités et les grands organismes de recherche.

En conséquence, l'expertise informatique a été mobilisée à différents titres :

- trois enseignants-chercheurs de l'Université de Montpellier ont été chargés de concevoir, de mettre en place et d'assurer le suivi de la formation ISN des professeurs des lycées, dans le cadre des moyens arrêtés par l'autorité académique,
- ces trois enseignants-chercheurs ont été systématiquement invités à participer aux commissions de sélection et aux jurys d'habilitation organisés par l'autorité académique, avec la participation répétée d'un chercheur d'un grand organisme (Inria),
- ces mêmes enseignants-chercheurs ont été fréquemment invités à l'accompagner en visite de classes ISN encadrées par des professeurs déjà habilités ayant suivi le module ISN1, avec l'accord de ces derniers et celui du proviseur de l'établissement, visites ayant généralement pour objectif la *validation définitive* du professeur visité, le rôle des enseignants-chercheurs étant fixé à la seule attestation de la qualité informatique des contenus exposés¹⁹.

En octobre 2013, le recteur de l'académie a pris sa retraite, mettant fin à ce que nous avons appelé la phase 1. Le changement de recteur s'est accompagné d'une modification de la responsabilité académique de l'ISN : deux IA-IPR, respectivement issus des disciplines SVT et STI, ont remplacé l'IA-IPR de Mathématiques.

L'année 2013/2014 (phase 2) a été caractérisée par :

19. Pour mémoire, seuls des avis « informatiques » favorables, voire très favorables ont été exprimés.

- le déroulement des deux modules de formation ISN1 et ISN2 annoncés et mis en place, comme de coutume, au cours de l'année précédente,
- la tenue des jurys d'habilitation hors la participation de l'équipe pédagogique (non sollicitée à l'issue du module ISN1, contrairement à l'usage adopté en phase 1),
- l'absence de sollicitation des membres de cette équipe à épauler « informatiquement » les inspections destinées aux habilitations définitives,
- la décision des autorités académiques de modifier les contenus des modules ISN1 et ISN2 pour l'année suivante.

L'année 2014/2015 (phase 3) a vu :

- pour l'équipe pédagogique, l'obligation de refondre les modules ISN1 et ISN2 en un unique nouveau module ISN1 augmenté (5 journées en présentiel au lieu de 3),
- pour l'autorité académique, la reprise en main du module ISN2 sous sa responsabilité exclusive (2 journées en présentiel au lieu de 3).

Sentiments. Au cours de la phase 1, la constitution et le fonctionnement des jurys d'habilitation étaient explicités au cours du module ISN1 :

- des équipes d'IA-IPR et d'informaticiens pour les jurys,
- un entretien individuel d'une demi-heure portant pour moitié sur une présentation du projet informatique (voir le contenu du module ISN1) et pour moitié sur le projet pédagogique du candidat pour l'encadrement d'un groupe de lycéens en spécialité ISN.

Il nous semble que ces jurys ont fonctionné à la satisfaction générale car, à deux ou trois exceptions près, tous les collègues qui se sont présentés à l'entretien ont été habilités.

Suite au changement de responsables académiques, l'habilitation est devenue du seul ressort des autorités académiques²⁰. Nous avons estimé cette mise à l'écart de l'équipe pédagogique contre nature.

En cette quatrième année, la refonte des modules ISN a réduit notre intervention (5 journées en présentiel au lieu de 2×3) et, par là, forcé à l'abandon de certains des thèmes présentés dans la suite initiale ISN1+ISN2 (cf. les contenus détaillés en annexe).

Mais surtout cette refonte a supprimé la « respiration » qui existait jusqu'alors entre les modules ISN1 et ISN2, une pause marquée par l'habilitation, et que nous pensions productive par la prise de recul qu'elle autorisait aux participants, un temps de maturation des connaissances déjà acquises par eux avant d'attaquer le module ISN2.

20. Sous des modalités que nous ignorons, comme nous ignorons quelle expertise informatique a remplacé celle de notre équipe pédagogique.

Ce dernier point nous paraissait particulièrement important. La session ISN1 « nouveau modèle » qui a constitué la phase 3 a confirmé nos réserves.

Enfin, il a été indiqué plus haut que sur 180 participants ayant suivi le module ISN1, 80 avaient suivi le module ISN2. Au-delà d'une interprétation de ces chiffres par une dérive « naturelle » entre les deux modules, il convient d'avoir à l'esprit que :

(1) la cinquième et dernière session d'ISN1 regroupait (bien qu'en partie seulement) les deux anciens modules ISN1 et ISN2, et ses participants n'étaient pas justiciables d'un module ISN2 « à l'ancienne » (le seul pris en compte dans cette estimation de suivi),

(2) des participants aux quatre premières sessions d'ISN1 ont exprimé leur souhait de suivre une session ISN2 dont le contenu leur avait été présenté et brièvement commenté (cf. les contenus détaillés en annexe), mais n'ont pu obtenir satisfaction du fait de la disparition de ce module au profit d'un « nouvel ISN2 »²¹.

Conclusion

Ce rapport rend compte d'une expérience de quatre années de « formation ISN » de professeurs du secondaire volontaires pour encadrer des groupes d'élèves de terminale S en spécialité *Informatique et sciences du numérique*²².

Or cette situation ne saurait être pérenne et, vu les enjeux pour les jeunes générations, l'objectif s'exprime bel et bien comme l'écrit le *Communiqué de la Société informatique de France* suite aux annonces de François Hollande et Najat Vallaud-Belkacem²³ :

« Dans un esprit de cohérence, nous attendons maintenant avec impatience la prochaine annonce politique qui permettra de disposer de professeurs d'informatique dont les compétences seront égales à celles que peuvent avoir les professeurs de mathématiques en mathématiques, de physique en physique, de français en français... »

Mais en attendant ? Comment participer au mieux à ce qui, dans l'immédiat pour nos autorités, tient lieu d'introduction de l'informatique dans les cursus du primaire et du secondaire ?

21. Dont nous n'avons eu connaissance ni du contenu ni des participants.

22. Ainsi qu'il rend compte d'une collaboration à géométrie variable entre responsables pédagogiques universitaires et autorités académiques locales.

23. <http://www.societe-informatique-de-france.fr/wp-content/uploads/2015/05/2015-05-Communique-SIF-annonces-NVB-et-FH.pdf>

La phase de transition risque d'être encore longue²⁴ et la question se pose d'élaborer collectivement des moyens destinés à aider au mieux les collègues du primaire et du secondaire chargés de la mise en œuvre progressive d'éléments d'informatique dans leur enseignement, telle qu'annoncée par les diverses autorités politiques et administratives, et ce, quoi qu'on pense de la méthode apparemment choisie²⁵.

Ce rapport se veut être une contribution au débat, et peut peut-être susciter d'autres comptes-rendus d'expérience²⁶.

ANNEXE

Niveau 1

Algorithmique

À partir de l'exemple d'un algorithme de multiplication (inspiré d'une méthode égyptienne vieille d'environ 4000 ans, également appelé *multiplication par décalage binaire*), sont présentés plus formellement les algorithmes dits « itératifs » :

- les instructions sont définies récursivement à la manière classique des *programmes tant que* : instructions élémentaires, instructions séquentielles, instructions répétitives,
- où les instructions élémentaires sont les actions de base dont sont dotés les processeurs auxquels l'algorithme est destiné (certains processeurs peuvent multiplier deux matrices, ou alors colorier un sommet d'un graphe, ou encore retourner une pile de crêpes),
- les instructions conditionnelles sont présentées et adoptées comme raccourcis de répétitives jugées trop peu « confortables » dans leur expression. Après le petit exercice classique : traduire un *si ... alors ...* sous forme d'un *tantque ... faire ...*.

24. Nous ne disposons pas des tableaux nécessaires à une estimation réaliste, mais disposer d'un corps « suffisant » d'agregés et de certifiés en informatique prendra à l'évidence « un certain temps ».

25. http://cache.media.education.gouv.fr/file/MEN_SPE_2/84/6/2015_BO_SPE_2_404846.pdf et http://cache.media.education.gouv.fr/file/CSP/91/2/prog_Informatique_et_creation_numerique_19_mai_425912.pdf

26. Rien ne montre aujourd'hui que le Ministère envisage, à court terme du moins, de travail de synthèse de l'expérimentation faite autour de la création de la spécialité ISN. En ce qui concerne l'académie de Montpellier, aucune enquête n'a été diligentée auprès des participants à cette formation, et aucune réunion de simple débriefing n'a été proposée à l'équipe pédagogique.

L'impérativité de la *spécification* des algorithmes est soulignée (doivent y figurer une rubrique *donnée* et une rubrique *résultat*), la notion de *justification* est illustrée, avec en particulier des exemples d'usage d'*invariants* (du simple commentaire à l'outil de preuve).

La thèse de Church-Turing est présentée.

Programmation

Il s'agit d'une initiation (traditionnelle) incluant les notions de :

- type, variable, expression,
- affectation (l'action de base du processeur),
- conditionnelle,
- répétitive (bornée et non bornée),
- sous-programme
 - fonction : abstraction de variables dans une expression,
 - procédure : abstraction de variables dans une instruction,
- tableau.

Le langage d'implémentation est Python²⁷ dont l'apprentissage n'est pas recherché pour lui-même : seuls les éléments jugés indispensables à ce stade d'initiation sont présentés.

Quelques règles de « bonne conduite » sont édictées :

- pas de variable globale, pas de *break* dans les répétitives, pas de multiplicité des *return* dans les fonctions,
- séparation des sous-programmes « métier » des sous-programmes d'E/S et des programmes de test,
- spécification impérative des sous-programmes,
- usage de commentaires « utiles », déclaration d'invariants si on peut.

Représentation de l'information

Sont exposés les premiers principes de la représentation des :

- entiers (écriture positionnelle et addition, base 2, taille fixe et additionneur modulo, implémentation en machine),
- réels (mantisse et exposant, la norme IEEE754, effets de la troncature et effets d'échelle),
- textes (codage des caractères imprimables ou de contrôle, quelques exemples de l'ASCII à l'Unicode),
- images (les pixels d'un écran, formats matriciels, effets de la compression, formats vectoriels),

27. Raisons principales de ce choix :

- la syntaxe du langage permet une mise en œuvre assez rapide de petits programmes,
- c'est le langage officiel d'apprentissage de la programmation en CPGE.

— sons (définition physique, mesure, échantillonnage, profondeur, formats simples ou compressés).

Structure des ordinateurs

La présentation suit un plan en 3 parties :

- la genèse des ordinateurs où l'on introduit progressivement et en suivant la voie historique les étapes qui ont conduit à la structure que l'on connaît actuellement,
- l'architecture de base des ordinateurs exposant les grands principes communs à toutes les réalisations et quelques évolutions marquantes,
- les architectures évoluées mettant en œuvre des techniques visant essentiellement à améliorer les performances.

Afin d'illustrer les concepts présentés, on définit une architecture de base d'un ordinateur de type « Von Neumann » comprenant :

- des cellules mémoire de 12 bits,
- un ensemble de 3 registres : accumulateur, compteur ordinal, registre d'instruction,
- une unité arithmétique et logique capable d'effectuer 6 opérations : addition, soustraction, ET, OU, OU-exclusif, décalage,
- un jeu très élémentaire de 16 instructions (codées sur 4 bits),
- une mémoire de 256 cellules (dont les adresses sont codées sur 8 bits).

Cette architecture ne comporte volontairement ni adressage indexé ou indirect, ni instructions d'entrée-sortie, ni instruction mettant en jeu plus d'un opérande. Les modifications d'adresses, par exemple pour le parcours de tableaux, se font par calcul sur le contenu de la cellule contenant l'instruction concernée. Des exemples de petits programmes très simples sont écrits en langage machine ; d'autres sont proposés à titre d'exercice.

Graphes et programmation

Outre fournir aux participants l'occasion de mettre en œuvre les notions d'algorithmique et de programmation vues jusqu'alors dans la session de formation, l'objectif de cette section est multiple :

- introduire la notion de graphe (présentée comme une structure de base pour la résolution de problèmes),
- exemplifier la notion de processeur non numérique (exemple d'action de base : colorer un sommet d'un graphe),
- exemplifier la notion de structure de données (ici, de façon alternative, matrices booléennes et tableaux de listes),
- initier aux algorithmes de descentes dans les graphes (descente non contrainte, descente en largeur, descente en profondeur).

De plus, une notion d'algorithme non déterministe est invoquée à l'occasion des algorithmes de descente.

Projet

Il s'agit de mettre les participants dans des conditions similaires à celles que vivront les élèves de terminale S en spécialité ISN :

- travail de groupe,
- travail en temps limité,
- rapport écrit,
- soutenance orale.

Les participants se constituent en binômes, ils choisissent un sujet parmi une liste qui en présente une quinzaine (ils peuvent soumettre à approbation préalable un sujet de leur cru), et disposent de trois ou quatre semaines (selon les sessions) pour rendre leur rapport (qui doit contenir en annexe le code rendant compte de l'état d'avancement du développement de leur application).

Le projet doit pouvoir faire l'objet d'une présentation orale.

Pour fixer les idées, quelques-uns des sujets proposés :

- le jeu de la vie,
- le tic-tac-toe,
- le Mastermind,
- aide à la cryptanalyse du chiffre de Vigenère,
- tris de tableaux par comparaisons et complexité théorique/effective,
- simulation d'une machine de Turing,
- construction de tablatures de guitare (proposé par des participants).

Niveau 2

Programmation objets

Cette introduction à la programmation objets a pour objectif d'aborder²⁸ les concepts de :

- classe, objet,
- envoi de message, encapsulation,
- héritage,
- polymorphisme, résolution dynamique,
- méthode abstraite, classe abstraite.

Le langage choisi pour la mise en œuvre est Java. La notion de typage statique est présentée (par opposition au typage dynamique de Python vu en ISN1).

28. Et également de s'y limiter, la proposition faite semblant suffisamment ambitieuse.

Programmation récursive

Présentation du paradigme de la programmation récursive en mettant l'accent sur la nécessité du contrôle de la finitude de la suite des appels récursifs engendrés. Seuls sont traités des cas où cette finitude peut être certifiée sans difficulté.

L'accent est mis sur le soin à apporter à la formulation des spécifications des fonctions/procédures récursives dans la perspective de pouvoir vérifier que :

- (1) les paramètres effectifs des appels récursifs :
 - satisfont les spécifications relatives aux données,
 - sont d'une « taille » (à définir pour chaque problème traité) strictement inférieure à ceux de l'appel environnant ;
- (2) le résultat de l'appel environnant satisfait ses spécifications sous réserve que les appels récursifs lancés satisfassent les leurs.

Systèmes d'exploitation

Cette section est une introduction aux fonctions d'un système d'exploitation, notamment celles de son noyau.

Le choix s'est porté sur l'aspect fonctionnel plutôt que sur la description d'un système particulier et de ses commandes. Cette approche vise à comprendre les relations entre les applications, leur environnement et l'ordinateur sur lequel elles s'exécutent.

Après un bref historique on y détaille les fonctions de :

- gestion de fichiers,
- gestion de processus,
- gestion de mémoire,
- gestion des communications.

Réseaux

La section réseaux explique le fonctionnement des réseaux de communication en adoptant un découpage fonctionnel à trois niveaux : transmission, routage (incluant la commutation de circuits, de messages et de paquets) et service. Quelques exemples historiques de réseaux de communication (comme les réseaux télégraphiques) sont présentés sous ce point de vue.

Web

Cette section s'intéresse à un des grands domaines d'application de l'informatique aujourd'hui. Elle présente les trois piliers du web :

- les langages HTML et CSS (permettant de présenter un *modèle de document*, un *langage formel*, à *balises* et le principe de *séparation de la structure logique d'un document de sa présentation*),
- le système d'identification de ressources URL (permettant de présenter le concept des *systèmes hypertextes*),

- le protocole de communication HTTP (permettant de présenter un exemple de *protocole de communication* de haut niveau et *sans état*).

Par ailleurs quelques aspects particuliers de la programmation web et quelques concepts des bases de données sont abordés.

Complexité des algorithmes et des problèmes

L'objectif de cette section est de présenter les notions :

- d'ordre de grandeur,
- d'évaluation des algorithmes : notion de complexité asymptotique au sens du plus mauvais cas (en se limitant à la complexité en temps),
- de classification des problèmes de décision : les classes \mathcal{P} et \mathcal{NP} ,
- de réduction polynomiale de problèmes : la classe \mathcal{NPC} et le théorème de Cook-Levin.

La question \mathcal{P} vs \mathcal{NP} est présentée.

Contrôle de l'information

Cette section aborde les questions relatives au contrôle de la transmission et du stockage des données informatiques. Il ne s'agit pas de l'organisation, qui est du domaine des bases de données. Elle se décline en :

- intégrité : détection et correction d'erreur,
- compression : optimisation du volume des données textuelles et perceptives (son, image, vidéo),
- cryptage : confidentialité et authentification.

Les parties « intégrité » et « compression » s'appuient essentiellement sur la théorie de l'information. La partie « cryptage » s'appuie aussi sur la complexité.

Projet

Les participants ont passé l'« épreuve du projet » lors de la session ISN1. L'objectif n'est plus ici de la leur faire repasser mais plutôt de leur fournir l'occasion de mettre en œuvre, à propos d'une réalisation de taille « raisonnable », les acquis de la session ISN2.

Un sujet de projet commun est proposé (un jeu de plateau) selon les modalités suivantes :

- travail en groupe ou isolé,
- seules les sources sont à produire (des sources présentées de façon « exploitable » par le lecteur),
- l'application doit être réalisée en programmation objets (Java),
- le temps imparti est de plusieurs semaines (voire extensible).



Organisation du travail à l'âge informatique

Laurent Bloch¹

Mesurer, contrôler, payer le travail

La première révolution industrielle de la fin du XVIII^e siècle et surtout la seconde, de la fin du XIX^e (électricité et moteur à combustion interne), ont fait du travail une marchandise que des entreprises achètent à des travailleurs et dont il faut déterminer le prix.

Les économistes du XIX^e siècle ont beaucoup réfléchi à la valeur et au prix des marchandises. La valeur aurait pu être mesurée par l'utilité de la marchandise, mais l'utilité n'était pas la même pour tout le monde, et de toute façon on ne savait pas la mesurer. Alors, de Ricardo à Marx, tous sont tombés à peu près d'accord pour conclure que la valeur d'une marchandise était déterminée par la quantité de travail nécessaire à sa production, quantité elle-même mesurée par le temps de travail. Le temps de travail était facile à mesurer dans la grande entreprise de la seconde révolution industrielle (1875-1975) : horloges pointeuses à l'entrée des ateliers, contre-maîtres sévères et chronomètres tayloriens. Ce mode de production a atteint son apogée en France dans les années 1970, avec 40 % de la population active dans l'industrie, contre 20 % aujourd'hui².

Depuis quarante ans nous sommes entrés dans la troisième révolution industrielle, avec la généralisation de l'informatique permise par le microprocesseur à bas coût et la propagation du logiciel et de l'Internet. Sous ce régime toutes les tâches répétitives ont vocation à être automatisées, et le travail humain se concentrera dans les activités

1. lb@laurentbloch.org

2. <http://michelvolle.blogspot.fr/2011/05/le-nouveau-monde.html> Le blog de Michel Volle est une source particulièrement riche d'informations sur ces sujets.

mentales, intellectuelles ou psychologiques telles que la conception, l'ingénierie, les services, l'enseignement, les soins, la maintenance, etc. Les marchandises seront des assemblages complexes de biens et de services, élaborés par la coopération d'acteurs multiples dispersés dans le vaste monde, puisqu'aussi bien cette révolution cyberindustrielle³ a permis la généralisation du conteneur, qui a divisé les coûts de transport par 50⁴.

Ainsi que le note Pascal Lamy dans l'article cité en référence, cette nouvelle ère économique remet en cause nos instruments de mesure de la production et du commerce : lorsqu'un téléphone portable franchit une douzaine de frontières dans l'un et l'autre sens avant d'arriver dans la vitrine du vendeur, que valent les statistiques du commerce extérieur ?

Il en va de même pour le travail : l'horloge pointeuse échoue à déterminer si le salarié assis devant son ordinateur crée de la valeur pour l'entreprise ou s'il bavarde sur Facebook, et les logiciels de surveillance ne serviront qu'à stimuler l'inventivité de ceux qui voudront les déjouer. Le temps de travail n'est plus une mesure pertinente de la valeur des biens et des services.

Notons que si la révolution cyberindustrielle nous prive de certains procédés de mesure de l'économie, elle nous en procure de nouveaux. Nous avons écrit ci-dessus que l'économie (classique, néo-classique, marxiste...) ne savait pas mesurer l'utilité d'un bien, et que de toute façon il aurait fallu pouvoir le faire pour chaque individu en particulier : ce problème est désormais résolu⁵. Amazon, Google, Uber savent parfaitement, en analysant mes navigations sur le Web et mes achats passés, déterminer les emplettes ou les trajets qui peuvent m'être suggérés. Et toutes les compagnies aériennes pratiquent le *yield management*, qui consiste à déterminer, à chaque date, pour chaque candidat au voyage, quel est le prix maximum qu'il est prêt à payer.

Nature du travail informatique

L'informatique transforme donc du tout au tout le travail, mais qu'en est-il du travail informatique proprement dit ?

Les premiers ordinateurs, qui entrèrent en fonction à l'extrême fin des années 1940 et durant les années 1950, étaient consacrés à des travaux militaires ou scientifiques puisque, à cette époque, on pensait qu'ils seraient utiles essentiellement aux calculs des physiciens. Le projet phare de ces années fut SAGE (*Semi-Automatic Ground Environment*), système de traitement des données issues des radars de défense anti-aérienne, destiné à prémunir les États-Unis contre une attaque nucléaire

3. Cf. Laurent Bloch, *Révolution cyberindustrielle en France*, 2015, Economica.

4. Cf. Pascal Lamy, « Mondialisation : renouvellement conceptuel et renouvellement des politiques », *Commentaire*, mars 2012. Vol. 35, n° 137, pp. 118-126.

5. Cf. l'analyse de Michel Volle, Vincent Lorphelin et Christian Saint-Étienne parue dans *Le Monde* du 20 juin 2015, *Le triomphe de l'économie de l'utilité*, en ligne ici : <http://www.iceconomie.org/wp-content/uploads/2015/06/le-monde-20-juin-2015.pdf>

soviétique, dont le coût final atteignit huit milliards de dollars de l'époque. D'autres projets militaires d'une ampleur comparable permirent la naissance et l'expansion d'une population de professionnels de la programmation des ordinateurs et, au bout du compte, l'apparition d'une véritable industrie du logiciel, bien décrite par Martin Campbell-Kelly⁶. Cet afflux de financements militaires joua un rôle de premier plan dans l'élan initial de l'industrie informatique américaine.

Il ne fait aucun doute que lors de ces premières réalisations informatiques la part du logiciel fut considérablement sous-estimée tant sur le plan financier que sur ceux de la durée et de la complexité du travail à accomplir. On s'imaginait en être quitte avec quelques dizaines de lignes de langage machine écrites sur un coin de table pour faire marcher les ordinateurs qui, eux, exhibaient les signes extérieurs de leur prix élevé en pesant des dizaines de tonnes et en consommant des centaines de kilowatts-heure d'électricité. Aussi les dépassements de coûts et de délais furent-ils la règle et engendrèrent-ils frustrations et déceptions ; c'est d'ailleurs encore le cas aujourd'hui.

Les premiers programmeurs étaient des scientifiques recrutés pour leurs compétences en analyse numérique et en algèbre linéaire, puisqu'il s'agissait essentiellement, croyait-on, de résoudre des équations différentielles et d'inverser des matrices.

C'est du milieu des années 1960, avec le lancement de la gamme IBM 360, que l'on peut dater la généralisation de l'informatique pour la gestion des entreprises, et du milieu des années 1970 la naissance d'une véritable industrie du logiciel, essentiellement tournée vers la gestion avant qu'elle n'enfourche le destrier de la micro-informatique des années 1980 pour peupler le monde de traitements de texte et de tableurs. Encore une douzaine d'années et nos écrans seront pleins des navigateurs de la seconde moitié des années 1990.

Dès lors il fallut bien se demander comment faire travailler les informaticiens, qui se comptaient par centaines de mille, puis par millions.

Le salut par Taylor ?

La grande industrie de la seconde révolution industrielle avait trouvé une doctrine avec Frederick Taylor et son organisation scientifique du travail⁷. Les managers de l'informatique, qui n'avaient pas conscience de vivre une révolution et même d'en être les acteurs, ont cherché à en transposer les idées dans leur activité.

Tout travail manuel, m'a appris un ami établi en usine après des études de philosophie et d'histoire et avant une reconversion à l'informatique, tout travail manuel, donc, repose sur une combinaison de contrainte physique et de conviction morale. C'est le fond du taylorisme, qui mise beaucoup sur la contrainte physique et peu sur la conviction morale. L'exemple extrême en est le travail militaire, où il faut donner

6. Martin Campbell-Kelly. *Une histoire de l'industrie du logiciel*. MIT Press (traduction française de Pierre-Éric Mounier-Kuhn), Cambridge, Massachusetts (Paris), 2003.

7. *The Principles of Scientific Management*, 1911.

sa vie : le texte extraordinaire du colonel Ardant du Picq explique bien comment ce sacrifice peut s'obtenir⁸.

Contre toute vraisemblance et tout espoir, les managers ont misé sur la contrainte physique pour assurer la productivité des informaticiens : ateliers organisés comme les chaînes des usines automobiles, où chaque programmeur devait écrire ses 50 ou 100 lignes de code, puis les passer au suivant... Sous des formes moins caricaturales le rêve de l'usine informatique est encore bien vivant⁹, c'est pourquoi nous allons examiner quelques façons curieuses d'envisager le travail informatique, afin de mettre au jour des erreurs qui s'y glissent et qui entraînent des échecs coûteux. Le lecteur qui vit dans le monde de la recherche pourrait croire ces exemples périmés : qu'il se détrompe, en beaucoup d'endroits ces méthodes aberrantes font encore florès.

Conception et réalisation

Assez logiquement, cette vision de la production informatique empruntait aussi à l'industrie automobile son modèle de division du travail : des ingénieurs conçoivent et dessinent des plans généraux, que des dessinateurs transforment en dessins détaillés, les ouvriers professionnels règlent les tours et les fraiseuses, les OS alimentent les machines. De même, des analystes concepteurs coucheraient sur le papier les spécifications du logiciel, avec un niveau de détail de plus en plus fin au fur et à mesure que l'on descendrait dans la hiérarchie, puis des programmeurs, ravalés au simple rang de codeurs, n'auraient plus qu'à écrire le programme, parfaitement déterminé par les spécifications.

Bien entendu, non seulement ce modèle a disparu de l'industrie automobile, mais il n'a jamais fonctionné dans l'industrie du logiciel. Plus exactement, il a été mis en pratique avec des résultats catastrophiques, en l'occurrence les millions de lignes de programmes Cobol dont personne ne sait plus depuis longtemps comment ils marchent ni même ce qu'ils sont censés faire exactement, mais qui règlent encore le fonctionnement de nos comptes bancaires et de nos contrats d'assurance.

Une vision managériale périmée : le cycle en V

Le cycle de développement « en V » hérité des métiers du bâtiment est encore préconisé par les méthodes traditionnelles. La figure 1 reproduit le schéma traditionnel de ce cycle, qui se parcourt en commençant en haut à gauche par la spécification des choses à faire, et en terminant en haut à droite par la recette du projet. Entre-temps on aura procédé, par étapes aussi bien séparées que possible, à la conception globale, puis à la conception détaillée, enfin au « codage ».

8. *Études sur le combat*, 1880, <http://gallica.bnf.fr/ark:/12148/bpt6k864841>

9. De mon expérience de DSI j'ai tiré quelques procédés de diagnostic : dès que j'entends les locutions « outil informatique » ou « usine informatique » je sais que le cas de mon interlocuteur est sans espoir.

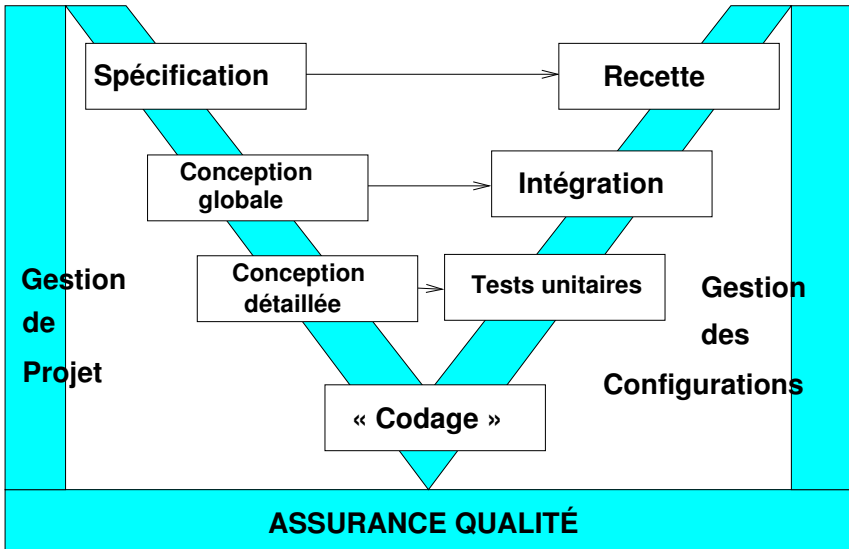


FIGURE 1. Cycle de développement « en V » du logiciel selon la vieille école.

Cette façon de faire est en voie d'abandon au profit des « méthodes agiles », plus à la mode et dont nous parlerons ci-dessous. Comment se propagent, puis reculent et disparaissent les méthodes de travail ? Les sociétés de services et les cabinets de consultants jouent un rôle capital dans ce processus, en effet ce sont eux qui implantent les méthodes nouvelles, d'abord chez leurs clients les plus riches et les plus dépensiers, puis de proche en proche jusqu'aux PME. Ce sont eux par exemple qui ont instauré la mode ruineuse des logiciels intégrés tels SAP et Oracle E-Business Suite, avant de leur substituer aujourd'hui les systèmes en nuage (*cloud computing*). Il faut en effet savoir que si dans les années 1960 les entreprises qui s'informatisaient savaient qu'elles allaient devoir écrire des programmes et donc se doter de compétences adaptées, il y a quelques décennies que la mode est à l'externalisation, dont les managers espèrent souvent qu'elle leur permettra de renoncer à toute compétence informatique à l'intérieur de l'entreprise.

Tribut à Frederick P. Brooks Jr.

Frederick P. Brooks Jr fut au début des années 1960 un des principaux concepteurs de l'OS 360, le système d'exploitation des *mainframes* IBM, et le chef de projet de sa réalisation. Il convient de signaler qu'en 2015 le descendant de ce système est toujours utilisé à grande échelle et repose toujours sur la même conception. Il s'agit

donc de l'un des plus gros et des plus longs projets de l'histoire de l'informatique, ce qui n'empêche pas son auteur de jeter un regard (auto-)critique sur la plus ambitieuse entreprise d'ingénierie des soixante dernières années : l'écriture des systèmes d'exploitation.

En 1975 Brooks a écrit un livre intitulé *Le mythe de l'homme-mois*¹⁰, où il tire les leçons d'une vaste expérience qui, de simple programmeur, l'avait conduit à la tête d'un gigantesque projet. En 1995 il en a publié une « édition du vingtième anniversaire », considérablement augmentée de réponses aux objections que lui avait values l'édition initiale, et de nouveaux développements inspirés par l'évolution technique, sociale et intellectuelle de l'informatique pendant cette période.

Quiconque possède la moindre expérience du développement informatique ne peut qu'être frappé par la profondeur et la finesse des observations et des analyses de Brooks, dont la plupart n'ont pas pris une ride malgré les bouleversements du monde informatique. Ce praticien n'a jamais l'œil aussi sûr que pour analyser ses propres erreurs, et l'on constate en le lisant que l'on peut à la fois être un homme de terrain et posséder une vaste culture. Au-delà de quelques phrases lapidaires qui ont frappé ses contemporains (« si une femme peut faire un enfant en neuf mois, neuf femmes peuvent-elles le faire en un mois ? », « lorsqu'un projet a pris du retard, ajouter du personnel à l'équipe ne fait qu'augmenter le retard »), il se livre à une analyse détaillée de l'activité de programmation, de sa nature intellectuelle et de la manière de mener un projet, analyse nourrie de multiples exemples concrets. Il est le précurseur des méthodes modernes que nous évoquerons ci-dessous, et il n'a jamais cru à des inepties comme le cycle de développement en V. En ces temps où les informaticiens réels sortent du tunnel de trente ans de gestion de projet et de dévaluation systématique de leur activité par des managers dont la morgue n'égale que l'incompétence, il est salubre de relire ce livre pour oser penser que ces méthodes qui leur ont été imposées étaient non seulement désagréables mais ineptes, ce qu'a démontré à l'envi une avalanche de catastrophes dont seul le secret a pu dissimuler l'ampleur. Nous ne saurions bien sûr nous dissimuler que les informaticiens ne sont pas totalement innocents de leurs malheurs, et qu'aux temps de leur splendeur ils ont abusé sans vergogne de leur pouvoir, créant par là les conditions d'un retour du bâton.

Un point intéressant à noter, c'est que le projet qui constitue la principale expérience de Brooks, l'OS 360, fut aussi un grand chantier d'application des méthodes industrielles à la réalisation de logiciel. Que l'un des principaux responsables du projet en vienne à écrire un livre, à la dimension autocritique explicite et assumée, qui analyse de façon détaillée les raisons pour lesquelles ces méthodes ne conviennent

10. Frederick P. Brooks, Jr. *The Mythical Man-Month* (traduction de Frédéric Mora : *Le mythe de l'homme-mois*). Addison-Wesley (Vuibert pour la traduction), Reading, Massachusetts (Paris), 1975-1995.

pas, me semble une contribution de poids à la critique de la taylorisation informatique. Les universitaires ne sont pas très bien placés pour l'observer, mais cette approche taylorienne est encore très en vogue, je l'ai rencontrée il y a moins de cinq ans.

J'ignore s'il a été le premier, mais en tout cas Brooks fut un des premiers à proclamer clairement qu'il est illusoire de prétendre établir au début d'un projet un cahier des charges et des spécifications immuable dont la maîtrise d'œuvre devra assurer docilement la réalisation fidèle. Il explique que le plus difficile dans une telle entreprise consiste à définir le but à atteindre, et que l'on ne peut y arriver que par itérations successives : réalisation d'un prototype sommaire, que l'on montre au donneur d'ordres, qui fait part de ses critiques et de ses suggestions à partir desquelles sera réalisé un second prototype plus élaboré, et ainsi de suite, pendant tout le cycle de vie du système, en fait.

En praticien consommé et en observateur assidu, Brooks a noté que dès lors que l'écriture d'un programme n'est pas effectuée par une personne unique qui en a eu l'idée, mais qu'un groupe de demandeurs s'adresse à un groupe de réalisateurs, les trois quarts du temps cumulé qu'ils vont consacrer ensemble à définir et à construire le programme seront consacrés à des échanges d'information entre eux. C'est notamment pour cela qu'ajouter du personnel à l'équipe d'un projet en retard ne fait qu'accroître le retard : il faut former et informer les nouveaux arrivants, et cela prend du temps qui s'ajoute à celui que les développeurs consacraient déjà à la communication au sein de l'équipe, beaucoup plus de temps que pour l'équipe de balayeurs de couloir auxquels certains managers aiment bien comparer les programmeurs. « Les hommes et les mois ne sont interchangeables que lorsqu'une tâche peut être divisée entre plusieurs travailleurs *sans réclamer de communication entre eux.* »

En s'inspirant d'une idée de Harlan Mills, Brooks préconise une organisation de l'équipe de développement selon le modèle d'une équipe chirurgicale dans un bloc opératoire : de même que seul le chirurgien en titre manie effectivement le bistouri, dans une telle équipe seul le chef-programmeur écrit effectivement des lignes de code, ou si d'autres membres de l'équipe en écrivent, c'est sous son étroite responsabilité. Il est assisté d'un adjoint (le *copilote*), pourvu des mêmes compétences que lui, en général plus jeune, avec qui il s'est concerté pendant les phases préparatoires, qui assiste à toutes les phases de son travail, notamment dans les phases de conception, mais s'il écrit du code c'est sous la responsabilité du chef-programmeur. L'équipe du chef-programmeur est encore complétée par d'autres fonctions assurées par des spécialistes dont certains ne seront éventuellement pas employés à plein temps, qui pourront contribuer à plusieurs projets et dont voici l'effectif complet type :

- le chef-programmeur ;
- l'adjoint du chef-programmeur ;
- un administrateur qui s'occupe des tâches de gestion (éventuellement à temps partiel pour le projet) ;

- un rédacteur-documentaliste chargé de produire, maintenir à jour et publier les documents relatifs au projet ;
- un responsable des essais, qui prépare les jeux de tests, les exécute et assure les fonctions de recette ;
- un expert du langage de programmation dont le rôle est d'aider l'équipe du projet à résoudre les problèmes techniques de développement (éventuellement à temps partiel pour le projet) ;
- un responsable de l'outillage, c'est-à-dire de l'installation et de la configuration des logiciels utilitaires nécessaires au projet, tels que système de gestion de configuration ou système de gestion de versions ;
- un archiviste responsable de l'intégration qui assure l'assemblage des différents programmes réalisés par l'équipe ;
- deux secrétaires.

Je doute qu'une telle organisation ait été réellement mise en place pour beaucoup de projets informatiques, ce à quoi une collègue me rétorquait un jour qu'il n'y avait que pour l'informatique que cette méthode n'était pas employée, alors qu'il s'agissait du modèle standard d'organisation des équipes de conception et de développement dans tous les autres domaines d'ingénierie. Mais ne s'agit-il pas d'un trait archaïque des industries de fabrication ? Depuis la première édition du livre de Brooks sont apparus de multiples outils informatiques d'aide au développement qui remplacent au moins partiellement des assistants humains : gestionnaires de version, gestionnaires de configuration, outils de documentation et de publication, débogueurs graphiques, etc. Le Web et les forums en ligne se révèlent de formidables moyens d'assistance technique : je sélectionne avec ma souris le message d'erreur que vient de m'envoyer mon logiciel (compilateur ou autre), je le colle dans la fenêtre de *Google*, et bien souvent j'ai la réponse, voire des dizaines de réponses, avec en prime l'adresse électronique d'experts dont j'ignorais jusqu'à l'existence. Ce n'est pas une démarche intellectuelle très académique, mais elle a le mérite de l'efficacité, avec cette restriction qu'elle fonctionne surtout pour les logiciels libres. C'est ainsi que les informaticiens améliorent leur productivité, et non pas avec des méthodes tayloriennes.

Fort de ses années de pratique, Brooks nous livre une remarque qui pourrait sembler anecdotique, mais qui en réalité touche au cœur de la question : rédiger des cahiers des charges et des documents de spécification et les présenter à des comités de projet au cours d'interminables réunions, c'est extrêmement ennuyeux et démobilisateur. Écrire un programme qui tourne, même s'il ne fait presque rien, est une activité exaltante, non seulement pour le programmeur, mais aussi pour toute l'équipe et pour le maître d'ouvrage, qui se trouvent galvanisés de voir les choses avancer concrètement. Il faut donc réduire au minimum les activités démobilisatrices dans lesquelles le projet va s'enliser, et multiplier les occasions de s'enthousiasmer pour une activité créatrice, dans les limites des objectifs poursuivis bien entendu.

En effet, et c'est en réalité assez mystérieux, mais quiconque a programmé, ne serait-ce qu'un peu, le sait, réussir à faire marcher un programme procure une sensation intense d'allégresse, totalement disproportionnée à l'enjeu réel du programme, qui peut être un exercice minuscule, et en revanche aller se coucher tard dans la nuit sans avoir réussi à résoudre un problème de programmation laisse un arrière-goût de découragement tout aussi excessif. Même des psychologues de la programmation comme Gerald Weinberg n'ont pas vraiment élucidé ce phénomène.

Ce qui fait travailler les programmeurs

Brooks et Weinberg sont d'accord pour préconiser une « programmation sans ego » : tout acte de programmation est en fait un acte social, puisque tout programme a vocation à être généralisé et partagé. L'abnégation de l'auteur contribuera à rendre ce partage plus facile et plus large. Cette vision évoque une morale un peu naïve, mais on aurait tort de la négliger, elle est effectivement au cœur de l'acte du programmeur, et c'est elle qui explique en grande partie le développement considérable du mouvement du logiciel libre. Et c'est faute de l'avoir comprise que nombre de managers extérieurs au monde de la programmation ont fait échouer des projets informatiques pour lesquels ils croyaient avoir choisi de bonnes méthodes qui n'étaient en fait que des mécanismes autoritaires déguisés mais inefficaces et promptement déjoués par les programmeurs indignés. En termes plus directs, si l'on persécute les programmeurs avec Merise et ISO 9000, on s'expose à un sabotage bien mérité.

De façon plus générale, ce qui fait travailler les programmeurs, et avec eux de plus en plus de travailleurs de cette nouvelle économie que l'Institut de l'Économie¹¹ a nommée, justement, l'*économie*, c'est ce que Michel Volle appelle le *commerce de la considération*¹² :

« La considération est cruciale aujourd'hui parce que le travail est devenu essentiellement mental : une entreprise qui n'écoute ni ses concepteurs, ni sa ligne de service, ne peut pas connaître les besoins des clients ni réussir ses innovations.

Le cerveau d'un concepteur ne sera en effet productif que si celui-ci peut discuter ses idées avec les autres métiers de l'entreprise, avec les dirigeants. Un concepteur que l'on n'écoute pas a tôt fait de se renfermer dans sa coquille. Il y fera des choses qui l'intéressent, l'amuse ou répondent à une mode parmi les chercheurs, mais une bonne idée ne peut être féconde que si elle est adoptée et mise en pratique par l'entreprise. Le manque de considération envers les concepteurs stérilise la conception et inhibe l'innovation.

[...]

11. Cf. <http://www.économie.org/>

12. Cf. <http://michelvolle.blogspot.com/2009/12/pour-un-de-la-consideration.html>

La considération est un *commerce* : il ne s'agit ni de moralisme, ni de sentimentalisme, mais d'un *échange* et comme tout échange celui-ci doit être *équilibré*. »

Méthodes agiles : *eXtreme Programming*

La méthode *eXtreme Programming*¹³ (XP en abrégé) se situe dans la lignée des méthodes de développement rapide ou de prototypage rapide tout en s'en distinguant, on peut dire aussi qu'elle est l'héritière des idées de Brooks, que nous avons rappelés plus haut. Elle a été créée à la fin des années 1990 par Kent Beck, Ward Cunningham et Ron Jeffries. Les auteurs du livre cité en référence l'ont introduite en France au début de notre siècle. Cette méthode et d'autres assez semblables sont maintenant appelées « méthodes agiles » et jouissent d'un certain succès après qu'aient été essayés les échecs des méthodes plus rigides évoquées ci-dessus.

Un développement conforme à la méthode *eXtreme Programming* (nous dirons « développement XP ») obéit à un cycle en V qui paraît assez semblable à celui que nous avons critiqué plus haut, mais au lieu de partir d'un cahier des charges monumental et immuable et de parcourir une fois le cycle en deux, trois ou quatre ans, on part d'une définition d'objectif très générale et laconique, en fait des *scénarios* établis par le client, et on va parcourir de nombreuses fois le cycle au cours d'itérations dont chacune pourra durer un laps de temps aussi bref que deux ou quatre semaines. Chaque itération amènera une définition plus affinée de l'objectif et un accroissement des capacités du système, étant entendu que dès la fin de la première itération le système sera opérationnel, même s'il ne fait à peu près rien. La durée de l'itération doit être fixée au départ ; une fois qu'elle est fixée il est important de la respecter et de procéder à chaque fin de cycle aux opérations de réception des travaux accomplis pendant cette période. L'intérêt de la méthode, on le comprend, réside dans la définition progressive mais néanmoins rigoureuse de l'objectif, et dans la capacité à s'adapter aux changements de cap en cours de navigation.

eXtreme Programming repose sur un certain nombre de *pratiques XP* canoniques :

- la programmation en binôme : tout le travail de programmation est effectué à deux programmeurs par poste de travail, un qui écrit et l'autre qui regarde ; la composition des binômes varie fréquemment, programmeurs expérimentés et débutants échangent régulièrement leurs places ; ainsi tous les membres de l'équipe connaissent le code de toutes les parties du projet et les débutants profitent de l'expérience des experts ;
- les tests unitaires : toute écriture d'une partie de l'application commence par la conception et l'implémentation des tests qui permettront de la valider ;

13. Cf. *Gestion de projet eXtreme Programming*, Jean-Louis Bénard et al., Eyrolles, Paris, 2004.

- la conception simple : on ne cherchera pas à anticiper les complications futures éventuelles, chaque fragment de code est rédigé de la façon la plus simple possible compte tenu des spécifications du moment ;
- le remaniement : il s'agit de la mise en facteur commun et de la généralisation de parties du code qui correspondent à des motifs (*patterns*) récurrents ;
- la responsabilité collective du code, induite et facilitée par la programmation en binômes de composition variable ; on note ici une divergence avec l'équipe du chef-programmeur de Brooks : les temps ont changé et l'autorité du chef n'est plus acceptable, du moins si elle est explicite ;
- les règles de codage, destinées notamment à assurer l'homogénéité du code, d'autant plus nécessaire que chaque programmeur doit pouvoir intervenir sur chacune de ses parties ;
- l'intégration continue permet la livraison d'un système opérationnel à la fin de chaque itération du cycle de développement, dont on rappelle que la durée est fort brève (deux à six semaines sont des valeurs habituelles) ;
- le rythme durable : les « charrettes » hystériques pour livrer un projet en retard sont prohibées et remplacées par un rythme de travail régulier.

À ces préceptes il convient d'ajouter un style particulier de relation avec le client. Précisons tout de suite cette notion de client : idéalement ce sera le vrai client, celui qui paie, mais comme la méthode XP suppose sa présence assidue au sein de l'équipe de développement et sa participation active tant à la rédaction des scénarios qu'à la réalisation des tests de recette, dans la réalité il ne sera pas toujours possible de satisfaire à cette condition, et le rôle du client pourra par exemple être tenu par ce que l'on appelle en gestion de projet la maîtrise d'ouvrage déléguée¹⁴. La méthode précise les droits du client :

- disposer d'une vue d'ensemble du projet, d'un calendrier et d'un budget prévisionnels ;
- en avoir pour son argent ;
- juger des progrès sur un système opérationnel, selon les tests reproductibles qu'il aura rédigés (ou fait rédiger) ;
- pouvoir changer d'avis, modifier les spécifications et les priorités sans s'exposer à des augmentations de budget exorbitantes ;
- être informé des dépassements de délais suffisamment tôt pour pouvoir réduire le périmètre fonctionnel afin de retomber sur la date de livraison initialement prévue ;
- interrompre le projet à tout moment, et disposer néanmoins à cet instant d'un système utile et utilisable conforme aux sommes investies jusque-là.

XP suppose un travail d'équipe très fusionnel : tous les membres de l'équipe, y compris le client, doivent être dans le même bureau, où les postes de travail

14. Dans l'appellation « maîtrise d'ouvrage déléguée », c'est le client qui délègue, et pour bien faire il importe que cette MOAD fasse partie de son organisation.

seront disposés de telle sorte que chacun puisse voir l'écran de chacun (les manuels donnent le plan conseillé pour le bureau).

La méthode définit aussi des *rôles XP* :

- programmeur ;
- client ;
- testeur ;
- traqueur (celui qui tient le tableau d'avancement des travaux) ;
- manager ;
- coach.

Nous avons déjà défini le rôle du client ; s'il doit cumuler son rôle avec un autre, ce ne peut être que celui de testeur.

Le manager est en fait généralement extérieur à l'équipe proprement dite. Il est le supérieur hiérarchique des programmeurs et peut leur demander des comptes, notamment sur les calendriers et les budgets. S'il doit ajouter un autre rôle au sien, ce ne peut guère être que celui de traqueur.

Le coach, à l'inverse du manager, appartient pleinement à l'équipe, il en est même la cheville ouvrière, il veille à ce que chacun joue bien le rôle qui lui est dévolu et respecte les pratiques XP, il aide le client à rédiger des scénarios.

La littérature XP ajoute à ces principes un adage moins formel mais important néanmoins : le principe « diviser pour régner » est d'une grande efficacité face à un ennemi, il est contre-productif et totalement hors de propos vis-à-vis de sa propre équipe.

On l'aura compris, XP est moins une méthode de conception au sens habituel du terme qu'une méthode d'organisation du travail de conception. Les principes XP ne peuvent que séduire quiconque a une expérience de la programmation et du développement, parce qu'ils constituent des solutions rationnelles aux problèmes bien connus de cette activité, et parce qu'ils formulent implicitement une critique radicale des méthodes punitives et improductives du passé.

On peut craindre que la mise en pratique des principes XP, qui supposent abnégation (« programmation sans ego ») et rigueur sans faille, ne se heurte à la sociologie réelle des populations concernées, notamment dans un pays latin comme la France où les signes extérieurs du pouvoir et du prestige ont souvent au moins autant d'importance que la satisfaction intériorisée du devoir accompli.

Une objection souvent formulée à l'encontre des principes XP, c'est qu'ils ne seraient adaptés qu'à des projets de petite taille. Mais justement, une des conclusions auxquelles notre expérience nous a conduit, c'est que la grande taille de beaucoup de projets n'est pas vraiment justifiée par l'objectif poursuivi, et qu'il faut essayer de réduire la taille des projets dans toute la mesure du possible. C'est très souvent possible. Plaidons pour des logiciels moyens et la sobriété des projets.



Qu'est-ce que l'intermédiation algorithmique ?

Stéphane Grumbach ¹

La révolution numérique transforme le monde. Alors que les nouveaux services sont largement plébiscités par la population, le système politique rappelle régulièrement l'illégalité des innovations disruptives pour l'équilibre social. Les données, matière première des rouages du numérique, contribuent à dessiner de nouveaux rapports de pouvoir entre citoyens, entreprises et États. La mutation profonde de nos sociétés, qui est riche de promesses mais également non dénuée de menaces, est irréversible. Elle est portée par une vague de transformations radicales de l'accès aux ressources, dont le mécanisme de base est l'intermédiation algorithmique.

Introduction

Les conflits entre les grandes plateformes numériques et les territoires se multiplient un peu partout dans le monde. La société Uber, emblématique des disruptions sociales en cours, affiche un tel palmarès en la matière, qu'un verbe nouveau, « uberiser », est apparu dans le langage politique, pour désigner le basculement sur une plateforme globale d'activités autrefois assurées par des entreprises locales. De nombreux secteurs économiques sont ou pourraient être touchés par l'uberisation. L'objectif de cet article introductif est de montrer ce qui est à la base de la profonde mutation qui ébranle nos sociétés, à savoir une nouvelle manière d'intermédiaire reposant sur des algorithmes et éloignée du monde physique dans lequel elle opère.

1. Inria.

Les données, dont on comprend bien qu'elles sont à l'origine de la rupture des équilibres en place, font l'objet d'une couverture médiatique sensationnelle. L'expression « *big data* » a connu une ascension fulgurante à partir de 2011 et la publication du rapport de McKinsey², qui montrait l'incroyable potentiel économique de l'exploitation des données : 300 milliards de dollars annuellement pour la gestion du système de santé américain, 250 milliards d'euros annuellement pour l'administration publique en Europe.

La croissance exponentielle de la production de données donne le vertige et justifie l'intérêt actuel pour les données. Mais le vocabulaire et les concepts en vue changent rapidement, démontrant, s'il en était besoin, l'extrême rapidité des bouleversements technologiques et sociétaux. La focalisation sur le *big data* disparaîtra comme a disparu la focalisation sur les technologies de l'information et de la communication qui ont permis le développement d'infrastructures essentielles qui, parce qu'elles marchent sans faille, ne constituent plus un sujet. Quand les catégories des applications possibles des données se feront plus claires, elles concentreront l'attention.

Les informaticiens, qui ont développé les technologies qui ont permis l'émergence des nouveaux services numériques, sont donc à l'origine, initialement involontairement, d'une révolution qui les dépasse désormais complètement et affecte notre devenir commun. Cette révolution repose sur trois piliers : la connexion continue des humains et de plus en plus des objets au réseau, la numérisation progressive du monde dit réel dans sa dynamique, et la puissance du calcul qui s'étend rapidement.

L'intermédiation est une fonction essentielle de nos organisations. Elle permet de mettre en relation des personnes avec des biens ou des services sur les marchés bifaces, c'est-à-dire les marchés entre producteurs et consommateurs de services comme les chauffeurs et les passagers.

Qu'est-ce qui a changé ? Pour les organisations traditionnelles pas grand-chose, si ce n'est une utilisation des outils informatiques qui améliore leur fonctionnement mais sans le remettre en cause. Par contre, d'autres acteurs ont émergé, à partir de rien, et ils remettent en cause les équilibres existants. Bien souvent ils s'insinuent entre les producteurs de services et leurs clients. On dit qu'ils les désintermédièrent. C'est le cas pour les sociétés de taxis et pour les journaux par exemple.

Ce que les données ont permis, et tout particulièrement le flux de données, donnant une image du monde réel à un instant donné et dans sa dynamique, c'est la capacité de réaliser l'intermédiation d'une manière plus efficace que les acteurs traditionnels. Plus efficace, parce que profitant pleinement de données auxquelles les entreprises traditionnelles n'ont pas accès. Plus efficace, parce que libérée des contraintes diverses des entités de l'économie réelle. Plus efficace également grâce à la capacité

2. *Big Data, the next frontier for innovation, competition and productivity*, McKinsey, mai 2011.

de personnaliser les services taillés sur mesure pour leurs utilisateurs. De surcroît, les acteurs de l'intermédiation algorithmique développent des plateformes qui assurent de nouveaux services essentiels pour nos sociétés, désormais indispensables au fonctionnement de tous les secteurs.

L'intermédiation algorithmique s'insinuera partout où existe un marché biface. Faut-il s'en réjouir ? Certainement. Exploiter les données permet des services d'une performance et d'une profondeur renouvelées. L'accès à la connaissance en offre un bon exemple. Faut-il s'en inquiéter ? Certainement également. À court terme, la réorganisation des services créera une disruption profonde dans la société. À long terme, l'usage des données orientera profondément les choix de société. Les équilibres qui se mettront en place entre États, plateformes et citoyens, seront déterminants.

Ces évolutions ne peuvent être analysées sans les rapprocher des enjeux globaux que représente la question des ressources au niveau de la planète. L'intermédiation algorithmique permet en effet un usage renouvelé des ressources, comme le démontrent les offres de partage ou de *crowd sourcing* par exemple³. Mais les liens entre révolution numérique et ressources pourraient être plus profonds qu'il n'y paraît à première vue.

La suite de cet article est structurée de la manière suivante. Dans la prochaine section, nous rappelons les mécanismes économiques de l'intermédiation, et présentons dans la troisième section l'économie des plateformes. Certaines lois particulières de l'intermédiation algorithmique sont présentées dans la quatrième section. Nous discutons ensuite des implications quant aux ressources dans la cinquième section et de quelques perspectives plus politiques dans la sixième section.

L'intermédiation

Les données sociales, c'est-à-dire les données qui proviennent des activités humaines, par opposition aux larges volumes de données scientifiques en particulier, sont pour l'essentiel stockées et traitées sur de grandes plateformes dont l'une des fonctions principales est de permettre à leurs utilisateurs d'accéder à des services. Ces plateformes, qui jouent avant tout un rôle d'intermédiaire, sont au cœur des nouvelles formes d'organisations économiques et sociales portées par la révolution numérique.

L'intermédiation est une activité essentielle au fonctionnement de toute société. Les intermédiaires mettent en relation des personnes avec les biens ou les services dont elles ont besoin ou qui pourraient leur être utiles. Au niveau individuel, recourir à des soins, obtenir un prêt d'une banque, organiser un voyage et plus généralement acheter des biens sont autant d'activités courantes qui font appel à des intermédiaires. Ceux-ci disposent des connaissances nécessaires à la fois à l'identification

3. N. Colin et H. Verdier, *L'âge de la multitude : Entreprendre et gouverner après la révolution numérique*, Armand Colin, 2012.

des acteurs potentiels ainsi qu'à leur mise en relation au moment opportun. À un niveau plus global, le commerce, les échanges internationaux, l'exploitation des matières premières ou la production de biens manufacturés sont autant de secteurs qui reposent sur un ensemble d'organisations dont certaines sont entièrement dédiées à l'intermédiation, c'est-à-dire à la mise en relation.

Omniprésente à tous les niveaux de nos sociétés, l'intermédiation permet de mettre en relation des catégories d'acteurs complémentaires, qui ont besoin l'une de l'autre pour développer leurs activités, comme les clients et les commerçants, ou les patients et les médecins par exemple. On parle d'*économie biface*, c'est-à-dire une économie associant deux groupes d'acteurs, dont les activités sont rendues possibles par la mise en relation via un réseau⁴.

Les exemples de marchés bifaces abondent. La carte de paiement constitue une bonne illustration du modèle biface, avec d'une part des consommateurs et d'autre part des commerçants. Les services de recherche d'emploi, ou plus généralement de recherche de partenaires quel que soit le type d'échange entre les personnes, forment également une forme courante d'intermédiation. Les systèmes de vente en ligne entre particuliers constituent un autre type d'exemple. Et bien sûr, tous les systèmes de l'économie numérique, comme le moteur de recherche, le partage de vidéo, les réseaux sociaux par exemple, mais également les systèmes logiciels *open source*, ou les « *store* » comme ceux d'IOS et d'Android, associant développeurs et utilisateurs.

L'opération de mise en relation entre *fournisseurs et consommateurs* de biens ou de services est à la base de l'intermédiation. Cette mise en relation peut toutefois prendre de nombreuses formes différentes. Le fournisseur et le consommateur peuvent être amenés à se rencontrer directement, comme dans le cas des services aux personnes, ou au contraire à n'interagir qu'au travers du bien ou du service, comme dans le cas de l'épargne gérée par les banques. Par ailleurs, les opérateurs de l'intermédiation peuvent être impliqués à un niveau plus ou moins important dans l'échange entre les acteurs, ce qui rend parfois leur activité d'intermédiation moins explicite, en partie masquée par leur activité dans la production des biens ou services.

L'intermédiation est donc fondamentalement une opération de *traitement d'information*. C'est l'information dont dispose l'opérateur d'intermédiation qui lui permet d'identifier les correspondances possibles entre producteurs et consommateurs. L'opérateur doit donc connaître au mieux les acteurs en présence, tant les producteurs de biens ou de services que les consommateurs. La quantité de connaissances en sa possession et la qualité du traitement analytique dont il est capable sont fondamentales pour mettre en œuvre une mise en relation pertinente et efficace. La taille de l'opérateur d'intermédiation est déterminante, et sa puissance dépend directement du nombre de ses utilisateurs, ce qui conduit inéluctablement à l'émergence de géants.

4. J.C. Rochet and J. Tirole, Platform competition in two-sided markets, *Journal of the European Economic Association*, 2003. Voir également Alvin E. Roth, The art of designing markets, *Harvard Business Review* 85.10:118–126, 2007.

On comprend pourquoi la révolution numérique bouleverse tant ce secteur d'activité, qui dépend de manière si cruciale des données et qui est littéralement entraîné par la puissance de leur traitement algorithmique. Mais la maîtrise des données n'est pas la seule raison pour laquelle l'industrie numérique introduit une révolution si importante *dans* l'intermédiation, et *par* l'intermédiation.

L'intermédiation est indissociable de la complexité de nos sociétés, qu'elle rend possible par la maîtrise de l'information, le développement de processus complexes, le partage des tâches et de la connaissance, une forme de gouvernance et d'organisation distribuées entre des acteurs indépendants. Ce type d'organisation est emblématique de l'ère numérique avec ses réseaux décentralisés orchestrés par des algorithmes asynchrones. L'intermédiation est dans leur nature même, et c'est donc tout naturellement que les acteurs du numérique étendent leur emprise à tous les niveaux de nos sociétés au moyen des systèmes d'intermédiation algorithmique.

Si la banque et l'assurance constituent des exemples bien connus d'intermédiation, abondamment étudiés par les économistes, un spectre d'activités très large intègre une part d'intermédiation, parfois peu visible, dans ses services. C'est le cas en particulier de tous les secteurs d'activité intégrés dans un modèle biface, associant deux types d'acteurs complémentaires. Identifier ces secteurs d'activité est très important aujourd'hui, parce que partout où il y a intermédiation dans un espace biface, le potentiel de disruption du numérique est fort.

Les acteurs de l'intermédiation algorithmique ont en effet la capacité de « *dés-intermédier* » les acteurs traditionnels, c'est-à-dire de les dessaisir de leur lien avec leurs « clients », et par là même d'ébranler sérieusement leur modèle économique, voire même dans certains cas de rendre caduque leur raison d'être. De nombreux domaines rarement considérés sous l'angle de l'intermédiation, parmi lesquels des secteurs aussi différents que l'enseignement, la presse, la santé ou encore la fiscalité, méritent une attention particulière. Ils assurent clairement une mise en relation entre élèves et professeurs, journalistes et lecteurs, soignants et patients, ou encore contribuables et citoyens, et ils sont tous à des degrés divers frappés par les opérateurs numériques.

L'économie des plateformes

Les acteurs de l'intermédiation de l'ère numérique développent des écosystèmes ouverts qui permettent d'une part d'héberger leurs utilisateurs de base, et d'autre part de fournir une infrastructure pour accueillir les services développés par des tiers. C'est le modèle des *plateformes*, biface par nature, qui intermédient entre des utilisateurs et des services. On parle donc de *plateformes d'intermédiation*. Ces plateformes poursuivent deux objectifs complémentaires essentiels :

- assurer un lien direct avec leurs utilisateurs ;
- attirer les services tiers dans leur écosystème.

Le lien direct avec les utilisateurs est essentiel pour recueillir les données de ces derniers. Mais maintenir ce lien relève de la gageure tant la désintermédiation peut être rapidement menée, et ce autant pour les acteurs de l'économie traditionnelle que pour ceux de l'ère numérique. Une fois désintermédié, l'opérateur perd la capacité à recueillir les données personnelles de ces utilisateurs.

Les fonctionnalités étendues du moteur de recherche qui propose un affichage direct, au-dessus de la liste des liens hypertextes, des informations qui répondent à la requête, comme la météo, des horaires divers, ou encore des extraits d'encyclopédie, et qui apportent un vrai confort pour l'utilisateur, constituent un bon exemple du pouvoir de désintermédiation. En effet, cette présentation de l'information par le moteur désintermédié de fait les acteurs offrant ces informations. L'accès à leur site n'est plus nécessaire. Le débat a fait rage à propos du moteur Google News, qu'un certain nombre de pays dont l'Espagne ont banni pour leurs organes de presse.

La construction de l'écosystème au moyen d'une API sur laquelle les développeurs peuvent installer leurs services est essentielle également. Une plateforme est d'autant plus utilisée qu'elle offre de services, et réciproquement plus elle a d'utilisateurs, plus il est important d'y proposer ses services. Comme dans tous les marchés bifaces, une plateforme doit développer les deux faces en même temps, en résolvant le problème de la poule et de l'œuf.

Les plateformes d'intermédiation partagent donc globalement, quel que soit le secteur d'activité, cette offensive vers les deux faces que sont les utilisateurs et les fournisseurs de services. Elles partagent de surcroît un même mode d'organisation⁵, autour de trois grands axes :

- l'infrastructure, constituée par l'ensemble des services de base offerts par la plateforme aux utilisateurs ou aux services tiers ;
- les données, accumulées par la plateforme, qui combinent les traces sur les différents services ;
- le réseau d'utilisateurs, et leurs interactions.

Suivant les secteurs d'activité et l'état de maturité des plateformes, ces différents niveaux sont plus ou moins importants. Dans le cas des plateformes marchés, comme AirBnB ou Uber par exemple, le niveau réseau est déterminant. C'est le cas également pour des plateformes sociales comme Twitter par exemple. À l'inverse, Android, qui constitue également une plateforme sur laquelle de nombreux services sont développés, est avant tout une infrastructure.

Les données sont la matière première pour toutes ces plateformes qui les exploitent plus ou moins efficacement suivant leur état d'avancement et le type de données. Mais là aussi, les principes généraux dans le traitement des données sont

5. Sangeet Choudari, *Platform Scale: How an emerging business model helps startups build large empires with minimum investment*, Platform Thinking Labs, 2015.

les mêmes pour toutes les plateformes, et on comprend ce que les plateformes numériques ont de révolutionnaire par rapport aux modes d'organisation qui les ont précédées.

Le traitement des données repose sur les trois piliers suivants. Premièrement, la collecte de données primaires, c'est-à-dire qui existent préalablement et indépendamment de la plateforme, suivie de leur analyse et leur transformation en vue de la génération d'une première catégorie de services. Deuxièmement, la collecte des traces d'utilisation des services et leur transformation en vue de la génération de nouveaux services à partir de ces données secondaires. Troisièmement, la collecte des traces de fonctionnement global du système en vue de son amélioration continue.

Les données primaires sont de manière générale des données extérieures à la plateforme, comme celles constituées par les pages Web pour le moteur de recherche. Elles sont donc créées par des entités indépendantes de la plateforme sans effort de cette dernière pour leur génération. La plateforme doit par contre être en capacité de les collecter, de les stocker et de les traiter.

Deux modèles coexistent pour la collecte des données primaires, suivant que la plateforme héberge ou non ces données. Dans le cas du moteur de recherche par exemple, les données sont générées et stockées indépendamment de la plateforme, et sont récoltées en explorant le Web continûment et en chargeant les pages découvertes. L'algorithme, le *crawler*, suit les liens hypertextes entre les pages et copie l'intégralité de ce qu'il trouve.

Dans le cas des réseaux sociaux par contre, la plateforme offre des services d'hébergement des données, que les utilisateurs chargent eux-mêmes, pour profiter des services de stockage et de traitement de données ainsi que des services de réseau. Les données primaires sont alors sous le contrôle de la plateforme. Ce type d'organisation se généralise avec les plateformes de nuage dans lesquelles les utilisateurs disposent de capacités de stockage et de calcul extensibles à bas coût.

De nombreuses plateformes combinent ces deux types de données primaires, collectées à l'extérieur et hébergées en interne. Les services de cartographie constituent un bon exemple de cette juxtaposition de données sur les cartes, avec de nombreuses informations obtenues de grands opérateurs de services urbains par exemple, et des informations saisies par les utilisateurs individuels du service : photos, vitesse de déplacement, évaluation de restaurants, etc.

L'incontestable rupture tient à la granularité des données. Les plateformes disposent de données à des niveaux de granularité variables, associant le gros grain des données globales, comme les cartes, avec le grain fin des traces des utilisateurs. Cette association permet d'avoir une cartographie du monde à des échelles variables, dans laquelle on peut zoomer. Une telle combinaison de granularités à une si large échelle était impensable auparavant. Elle permet bien sûr des services radicalement nouveaux. Les systèmes de suivi des comportements de la population, comme ceux

mis en place pour les épidémies à partir du moteur de recherche, constituent un bon exemple de cette cartographie fine de l'activité.

Enfin les plateformes sont en mesure de s'adapter continûment en analysant leurs données de fonctionnement afin d'offrir le service le plus performant. Elles poussent au plus haut point les techniques de marketing qui permettent de tester et de comparer des approches différentes. Les techniques de l'*A/B testing* permettent de présenter une page Web sous deux versions légèrement différentes à des groupes d'utilisateurs similaires et de sélectionner, potentiellement en temps réel, celle qui donne le meilleur résultat. La campagne d'Obama pour l'élection de 2012 a fait un usage massif de telles techniques.

Les lois de la nouvelle économie

Quelques lois fondamentales, qui diffèrent parfois de celles de l'économie classique, s'appliquent aux plateformes d'intermédiation dans leur ensemble. Elles permettent de mieux appréhender leur immense succès, la puissance de leur disruption, et les perspectives radicalement nouvelles ouvertes pour la société.

Loi de l'économie biface numérique

Le modèle économique de l'intermédiation tire son efficacité d'un traitement différencié des deux faces intermédiées, dès le lancement et tout au long du fonctionnement du service d'intermédiation. Pour la carte de paiement, les consommateurs payent une contribution fixe annuelle et négligeable, alors que les commerçants payent une contribution proportionnelle au volume d'échanges. Tout est mis en œuvre pour développer l'activité et maximiser à terme le profit de l'intermédiaire gérant le moyen de paiement.

Le modèle économique diffère donc fondamentalement d'un modèle de production classique où le volume d'activité conduit à des économies d'échelle et à une baisse des prix. Dans le cas de l'intermédiation, les prix doivent être bas, voire nuls au début, pour assurer le décollage de l'opérateur, qui sera en mesure, quand sa position est confortable, d'augmenter la contribution des acteurs. Un équilibre doit être établi entre une stratégie orientée vers l'augmentation des revenus et une stratégie orientée vers le déploiement et l'élargissement du réseau. Dans le cas de la plateforme de vente en ligne Amazon par exemple, Jeff Bezos, son fondateur, justifie sa stratégie de favoriser l'extension du réseau plutôt que d'augmenter le revenu de la plateforme de vente, et sur la longue durée son actionnariat le soutient dans cette stratégie, malgré des revenus nuls voire négatifs. Deux conceptions s'affrontent entre le potentiel à long terme et le bénéfice à court terme.

Pour la majeure partie des systèmes offrant un service non marchand, comme le moteur de recherche ou les réseaux sociaux, l'utilisateur de base accède au service en échange non pas de paiement financier mais de ses données, qui transitent par la

plateforme et peuvent être exploitées commercialement par la plateforme suivant des règles qui peuvent être amenées à évoluer.

Si l'intermédiation a comme objectif premier le développement des échanges entre les deux faces, elle permet également les échanges à l'intérieur d'une même face, de manière coopérative ou au contraire de manière compétitive. La coopération est fréquente dans de nombreux types de jeux multi-joueurs par exemple, et constitue également la base des modèles de développement des logiciels *open source*. La compétition intervient entre les acteurs offrant ou cherchant les mêmes services dans un marché élargi par la plateforme. Ces deux modes d'interaction jouent un rôle essentiel dans les plateformes de service comme les véhicules avec chauffeur, où les chauffeurs, mais également les passagers, sont évalués et donc en compétition sur chaque face.

Loi des externalités

C'est en général dans ses externalités que repose l'intérêt d'une plateforme d'intermédiation. Les externalités économiques sont les conséquences indirectes de l'activité première sur d'autres acteurs. Elles peuvent être positives ou négatives sur l'environnement global. S'il est rare dans l'économie classique de ne pas avoir d'externalités, dans le monde numérique, on pourrait dire que les externalités forment l'essentiel. C'est le cas bien sûr pour le moteur de recherche, qui bouleverse initialement bien moins l'économie de la connaissance que celle de la publicité, son externalité première. Google, leader des moteurs, génère des revenus de près de 60 milliards de dollars en 2014 dans le secteur publicitaire.

Les données récoltées par les opérateurs de l'intermédiation sont d'une incroyable richesse. Elles révèlent essentiellement ce qui se passe dans le monde, au travers d'un flux continu d'information. Elles mettent leurs détenteurs en capacité de développer une connaissance en temps réel de toutes les interactions entre acteurs. On comprend donc comment, de ces immenses masses de données, on peut extraire de nouveaux services, indépendants du service initial, précisément sur les externalités de l'opérateur. Pour tous les systèmes offrant une intermédiation non monétisée (moteur de recherche, réseaux sociaux, etc.), le modèle économique repose sur les externalités, qui sont donc l'essentiel de l'activité. Mais il n'y a aucune raison qu'il en aille différemment des intermédiations monétisées comme celles des transports ou de l'hébergement, même si le revenu initial provient d'un prorata de la transaction.

Des plateformes comme Uber travaillent aujourd'hui étroitement avec les administrations des villes. Il est clair que de tels opérateurs seront rapidement en capacité d'offrir des services aujourd'hui assurés par les administrations, mais d'une manière beaucoup plus efficace, pour la simple raison de leur agilité et de l'exploitation à temps partiel du travail d'une large communauté d'utilisateurs devenus acteurs de la gestion de la ville. Cette nouvelle économie des ressources, en particulier humaines, mais pas seulement, est au cœur de la stratégie des plateformes.

Les exemples d'externalités abondent, en particulier pour les services exploitant le géo-positionnement. Android ou iOS disposent de capteurs bien plus nombreux et de traitements de données bien plus rapides que n'importe quelle administration pour prévoir les déplacements ou le trafic. Les plateformes de vente peuvent produire des statistiques économiques en temps réel, qui peuvent contredire les statistiques officielles, etc.

Loi d'abstraction dans le nuage

Les données récoltées par les plateformes génèrent une couche numérique qui forme une sorte de miroir du monde physique qui, comme tout miroir, permet de tout voir mais sans réelle interaction avec le monde physique. Les plateformes reposent bien sûr sur des infrastructures physiques formées par les centres de données et de calculs et les systèmes de communication, qui sont non négligeables⁶, mais sans connexion avec le monde intermédié. Il est donc possible de réaliser l'intermédiation entre deux acteurs sans aucune implication avec le monde physique dans lequel sont ces acteurs. Ainsi Uber intermédie entre chauffeurs et passagers dans des villes dans lesquelles Uber n'a pas d'autres activités que l'échange d'information numérique. Uber a par contre une représentation de la ville, des véhicules en mouvement, de l'évolution des écueils possibles, qui lui permet de satisfaire les demandes de ses utilisateurs.

Cette abstraction, dans le nuage, est nouvelle. Séparer l'intermédiation de la production des biens ou services du monde physique est le cœur de la révolution si disruptive pour nos sociétés. Il est en effet désormais possible d'offrir une intermédiation sur des territoires sur lesquels on n'a aucune présence physique. Pire encore, l'intermédiation dans le nuage déploie une efficacité dont les intermédiaires traditionnels de terrain sont incapables. Cette montée en abstraction de l'intermédiation n'est pas sans évoquer des mécanismes fondamentaux de l'économie comme l'abstraction de la monnaie dans les échanges.

Loi de concentration

Force est de constater que les plateformes d'intermédiation, qui pour ainsi dire n'existaient pas avant le tournant du siècle, ont connu une croissance extrêmement rapide et sont aujourd'hui très puissantes. Il n'y a pas lieu de s'en étonner, tant les services qu'elles proposent sont d'une qualité exceptionnelle. Leur adoption est massive dans la population mondiale, et leur poids économique de premier ordre.

Parmi les cinq premières capitalisations mondiales⁷, Apple (\$740 B), Google (\$367 B), Exxon (\$357 B), Berkshire Hathaway (\$354 B) et Microsoft (\$340 B), on compte aujourd'hui trois entreprises leader de l'intermédiation algorithmique.

6. B. Walsh, *The surprisingly large energy footprint of the digital economy*, globalwarming-sowhat.com, 2015.

7. <http://www.forbes.com/global2000>, September 22, 2015

Dans presque tous les secteurs d'activité, une plateforme domine le secteur, avec un nombre d'utilisateurs bien supérieur à ses concurrents. Cette domination est en partie due au fait que la puissance des plateformes suit essentiellement une loi à la Metcalfe, et dépend de manière quadratique du nombre de ses utilisateurs. Elle tient aussi au fait que les idiosyncrasies régionales ont une influence limitée sur l'activité. Ainsi Uber opère un service universel dans le transport qui dépend peu des conditions locales et peut se déployer sur des territoires en éliminant les opérateurs locaux.

La manière dont cette domination se met en place tient quant à elle au fonctionnement du capital risque, qui est investi largement dans un grand nombre de projets, et ne récupère sa mise que sur un tout petit nombre de gagnants, qui doivent donc être propulsés au premier rang, en éradiquant la concurrence.

La concentration n'est bien sûr pas seulement capitalistique. Le nombre d'utilisateurs des plateformes suit une loi de puissance. Il y a dans tous les secteurs économiques une plateforme dominante suivie de plateformes dont l'importance décroît jusqu'à une taille modeste où de nombreux concurrents s'affrontent. La concentration est également géographique. Peu de pays ont su faire émerger de grandes plateformes, et dans le premier d'entre eux, les États-Unis, la baie de San Francisco, Venise du monde moderne, héberge la majeure partie d'entre elles.

Une nouvelle économie des ressources

Deux révolutions majeures secouent le monde, la révolution numérique et la révolution climatique. Toutes deux, conséquences des activités humaines, suscitent des inquiétudes et commencent à induire des changements profonds dans nos organisations. La révolution numérique, rendue possible par le développement de nouvelles technologies, semble ouvrir de nouveaux possibles. À l'inverse, la révolution climatique, externalité négative du développement des activités humaines, principalement depuis la première révolution industrielle, semble au contraire condamner des pratiques. Ces deux révolutions contemporaines l'une de l'autre sont probablement moins indépendantes qu'il n'y paraît à première vue. Le numérique offre en effet de nouvelles possibilités d'organisation et d'utilisation des ressources⁸, et ce tout particulièrement grâce à l'intermédiation algorithmique. Mais au-delà, on peut également voir dans le numérique un *feedback* de résilience pour nos sociétés⁹.

La nouvelle économie repose avant tout sur une horizontalisation de la société au préjudice de son organisation verticale. Le déploiement de la couche horizontale résulte de la connexion désormais continue de chaque individu aux systèmes de communication et à la capacité pour chacun d'interagir avec d'autres au moyen des

8. J. Rifkin, *The zero marginal cost society: The internet of things, the collaborative commons, and the eclipse of capitalism*, Palgrave Macmillan, 2014.

9. S. Grumbach, O. Hamant, *Digital Revolution or Anthropocenic Feedback?*, 2015.

différents systèmes de réseaux dits sociaux. La différence avec les systèmes d'émission d'information de l'époque précédant l'apparition d'Internet est importante. On disposait alors seulement d'une communication point à point, via le téléphone en particulier, et d'une communication de masse par les médias réservée à une portion infime de la population en capacité d'émettre. Inéluctablement, du déploiement de l'horizontalité permise par la technologie, allait découler un affaiblissement de la verticalité, c'est-à-dire, de la hiérarchie de structures qui sépare un individu des organes de gouvernance.

Les plateformes d'intermédiation jouent un rôle déterminant dans cette mutation. Elles contribuent en effet directement à l'effondrement de la verticalité par la désintermédiation des acteurs intermédiaires. Le phénomène est assez simple. L'intermédiation algorithmique opère un double déplacement du pouvoir, d'une part vers le bas, vers les personnes, utilisateurs des services, et d'autre part vers le haut, vers les plateformes elles-mêmes, au détriment des nombreux opérateurs intermédiaires. Le déplacement du pouvoir vers le bas, l'*empowerment* des utilisateurs, est permis par la capacité à accéder à des services de plus grande abstraction, indépendants de l'environnement local. Le déplacement vers le haut est permis par la capacité de la plateforme de réguler l'intermédiation entre un nombre très large d'acteurs, récupérant ainsi les précieuses données et les potentialités associées.

Les acteurs intermédiaires historiques sont donc menacés de désintermédiation. L'exemple des sociétés de taxis, dont le modèle économique est à ce point mis à mal que leur survie économique paraît illusoire, illustre parfaitement ce phénomène. Au-delà du modèle économique, c'est la raison d'être de nombreux opérateurs qui s'effondre. Une proportion importante des organes de presse au niveau mondial a disparu dans la dernière décennie¹⁰. Les maisons d'édition sont remises en cause par la capacité d'éditer soi-même et d'accéder au réseau de distribution via une plateforme comme Amazon par exemple.

L'ensemble des acteurs opérant sur des marchés bifaces seront radicalement affectés par les opérateurs de l'intermédiation algorithmique. C'est-à-dire qu'ils ont des chances d'être complètement ou au moins partiellement désintermédiés par les plateformes numériques. Pourquoi ? Parce que les plateformes numériques offrent des services de qualité de plus en plus grande et de plus en plus personnalisés. Fox et Patterson¹¹ expliquent précisément le mécanisme de production d'un livre qui peut désormais être réalisé soi-même en accédant directement aux différents corps de métiers, ce qui était autrefois l'apanage de la maison d'édition. Les coûts de production sont d'autant réduits.

10. R.G. Kaiser, *The bad news about the news*, <http://www.brookings.edu/research/essays/2014/bad-news#>

11. A. Fox, D. Patterson, Do-It-Yourself Textbook Publishing, *Communications of the ACM*, Vol. 58, No. 2, <http://cacm.acm.org/magazines/2015/2/182651-do-it-yourself-textbook-publishing/fulltext>

Cette capacité de connexion directe avec les corps de métier impliqués dans la réalisation, la production, puis la diffusion d'un livre, illustre ce phénomène d'horizontalisation qui permet à chacun d'accéder directement, sans intermédiaire institutionnel, aux différents services. L'institution qui détenait cet ensemble de compétences en interne et constituait un passage obligé est remise en cause. Le monopole sur ce savoir-faire détenu par l'ensemble de la profession s'effondre. L'accès aux ressources, en l'occurrence le savoir-faire, est donc radicalement changé.

La force des plateformes d'intermédiation tient à la séparation, qui peut être totale, comme pour Google, ou seulement partielle, comme pour Amazon, du monde physique dans lequel elles opèrent. Leur capacité à désintermédier est forte parce que leurs services sont de plus haut niveau d'abstraction, et que par rapport aux acteurs qu'elles désintermédient, elles ne sont contraintes ni par la sauvegarde d'un modèle existant que les acteurs vont essayer de préserver au moins en partie, ni par des rigidités réglementaires ou organisationnelles par exemple.

On comprend l'étendue de cette révolution des lors que l'on considère le nombre de secteurs qui relèvent d'un marché biface, avec producteurs et consommateurs de services, soit toutes les associations du type médecins et patients, enseignants et étudiants, administrateurs et administrés, etc. Les institutions étatiques n'échapperont pas à cette transformation, même si leur inertie peut être plus importante à cause de leur plus faible dépendance à la dureté des lois du marché. L'Université a tout à gagner d'une telle évolution dans laquelle les étudiants pourront recourir à des enseignements produits hors de l'institution dans laquelle ils sont inscrits. Ce phénomène est en pleine expansion puisqu'on demande déjà à un jeune ingénieur de montrer son curriculum sur les plateformes (comme GitHub par exemple), et plus seulement son parcours académique *stricto sensu*.

En quoi l'intermédiation algorithmique peut-elle conduire à une nouvelle politique des ressources ? C'est grâce à cette interconnexion de l'ensemble des acteurs, et à la capacité de mise en relation indépendante des intérêts particuliers des acteurs, qu'un nouvel accès aux ressources est possible. Le développement des systèmes de *crowd sourcing*, et également ceux de *crowd funding*, démontre amplement ce potentiel. Mais d'autres facteurs fondamentaux sont également à l'œuvre.

Les plateformes d'intermédiation rendent possible la réversibilité des faces, c'est-à-dire la possibilité d'œuvrer sur une face et sur l'autre du marché, être tantôt consommateur, tantôt producteur. L'exemple du transport, avec les chauffeurs occasionnels remplaçant les taxis, illustre ce mouvement qui offre une souplesse extraordinaire et remet en cause l'organisation et le droit du travail. Risques et opportunités se cachent derrière cette tendance. Mais ce qui apparaît clairement, c'est une nouvelle gestion des ressources humaines, permise par la capacité à utiliser des ressources rendues disponibles de manière agile.

L'intermédiation entre les acteurs humains s'étendra rapidement aux acteurs non humains. L'évolution d'un environnement énergétique, basé sur un producteur fiable

et des zillions de consommateurs, vers un système reposant sur des zillions de producteurs non fiables et des zillions de consommateurs, nécessitera une intermédiation entre objets producteurs ou consommateurs. La gestion de l'énergie, ressource essentielle, relèvera également de l'intermédiation algorithmique. Il suffit de suivre les investissements des grandes plateformes dans l'internet des objets et la domotique pour s'en convaincre.

L'horizontalisation n'est toutefois pas simple à réaliser. Si sa capacité à exploiter de manière intensive les ressources, parce qu'elles deviennent accessibles, est évidente, encore faut-il pouvoir garantir la qualité et la fiabilité que le système vertical a pu mettre en place. C'est un élément déterminant du succès de cette mutation de nos sociétés.

Le remplacement des structures verticales par des structures horizontales nécessite en premier lieu de reconstruire de manière horizontale ce qui fait la force des organisations verticales : la confiance. Sans confiance, aucune société ne peut fonctionner, mais les modes de confiance varient d'une société à l'autre et sont amenés à évoluer¹². L'un des rôles de l'État est de créer cette confiance par des mécanismes normatifs *a priori* et des capacités de contrôle *a posteriori*. Or celle-ci peut également être construite par la multitude au moyen de mécanismes de recommandation et d'évaluation. C'est ainsi que les chauffeurs sont évalués par les passagers. Inversement, les passagers sont évalués par les chauffeurs également, conduisant ainsi à des interactions qui font systématiquement l'objet d'une évaluation symétrique.

C'est tout à fait remarquable que le covoiturage, vieux rêve utopique, soit devenu une réalité ordinaire, dont les ramifications se répandent rapidement étant donnée l'évidente économie qu'elles permettent.

Ce qui garantissait la qualité d'un chauffeur, son employeur au-dessus de lui, peut donc être assuré par le bas, par ses utilisateurs. Ce phénomène d'horizontalisation généralisée des mécanismes de confiance est à la base du développement de l'économie des plateformes. Ce n'est pas un hasard si les mécanismes d'évaluation, de jugement par les pairs, se répandent dans tous les secteurs de la société. Ils accompagnent l'horizontalisation.

Les plateformes d'intermédiation sont incontournables. Elles offrent aujourd'hui des services essentiels, aussi fondamentaux que le sont l'énergie, les télécommunications ou les transports par exemple. C'est sur les services d'intermédiation que de très nombreux aspects de nos vies vont être organisés. Les services offerts respectent les grands principes du service public que sont la continuité de service, l'égalité dans l'accès et la non discrimination des usagers, et l'évolution et l'adaptabilité, qui en France ont été fixés dans le droit dans les années 1930.

La grande différence entre ces nouveaux services et les anciens services essentiels tient à l'absence complète de l'État dans leur définition et leur mise en œuvre.

12. F. Fukuyama, *Trust: The social virtues and the creation of prosperity*, Free Press, New York, 1996.

L'État assiste en spectateur à leur émergence, tente de les réguler, mais avec désormais peu de prise. Si les services publics ont été développés avec une perspective du bien commun définie au niveau national, les grands opérateurs d'intermédiation ont aujourd'hui une perspective essentiellement commerciale qui ne manquera pas d'imposer une redéfinition de leur contours et des droits et devoirs respectifs de l'État et des entreprises, non seulement quant aux services offerts mais également aux données personnelles manipulées.

La légitimité politique

Les plateformes opèrent sur des territoires bien souvent sans rencontrer de barrières, parce que nombre d'États défendent un principe d'ouverture *a priori* de l'Internet, avec libre circulation des données. C'est une situation qui diffère fortement d'autres secteurs de l'économie pour un ensemble de raisons.

D'abord, dans la plupart des secteurs relevant de l'économie des biens matériels ou des services, il existe des barrières normatives, douanières ou fiscales, établies de longue date, qui permettent de contrôler les échanges entre le territoire et le monde extérieur. Ce n'est pas le cas pour les plateformes, à la fois parce que, le secteur étant nouveau et en évolution très rapide, le législateur n'a encore eu que peu de possibilités de développer une réponse juridique adaptée, mais également à cause du principe, assez naïf, de liberté *a priori* d'Internet.

Certains pays, parfois même seulement des villes, défendent à l'inverse un droit à une souveraineté numérique qui justifie la construction de frontières virtuelles bloquant les échanges avec certains sites extérieurs. Ces frontières ne sont pas nécessairement le signe d'un investissement faible dans le numérique. La Chine est à la fois la deuxième puissance numérique et le pays qui a développé l'une des frontières les plus importantes. Ces frontières sont surtout connues pour leur utilisation dans la censure de l'information à l'intérieur du pays, parce que cette censure s'oppose au principe de liberté, mais leur raison d'être est plus générale et relève bien plus de questions économiques et politiques fondamentales.

Les plateformes d'intermédiation ont un mode opératoire qui diffère radicalement des grandes multinationales de biens ou de services dans les secteurs économiques traditionnels. En effet, ces dernières pénètrent les territoires par le haut, elles doivent se mettre en conformité avec un cadre contraignant et, en cas de conflit, négocier avec les autorités du pays, de manière transparente ou secrète. À l'inverse, les plateformes pénètrent les territoires par le bas, c'est-à-dire en touchant directement les citoyens qui adoptent leur services sans restriction. Les contradictions qui se font jour entre les autorités et les plateformes, ou encore entre les utilisateurs et les plateformes, se posent *a posteriori*, alors que les plateformes ont déjà une présence importante sur le territoire et peuvent se prévaloir d'une adhésion de la population, d'une forme de plébiscite.

La longue série de conflits qui opposent les plateformes d'intermédiation aux territoires depuis une quinzaine d'années montre l'extrême complexité pour les territoires de faire prévaloir le respect d'un cadre légal, qui n'est pas simplement violé par les plateformes, mais de manière plus profonde ébranlé par l'évolution des techniques et des usages dont les plateformes sont le fer de lance. La résistance à de nombreux changements imposés par les plateformes ne relève donc bien souvent que d'une illusion conservatrice visant à faire accroire que l'on peut sauver le fonctionnement de secteurs industriels devenus caducs.

Au-delà des secteurs économiques en pleine restructuration, les plateformes remettent en cause des aspects plus fondamentaux de nos sociétés, comme la protection de la vie privée, l'intensité du renseignement d'État, la légitimité des décisions politiques, la gouvernance institutionnelle, etc., sur lesquelles il convient de réfléchir dans une perspective long terme, c'est-à-dire en prenant en compte l'irréversibilité et la profondeur des changements.

L'exemple du droit à l'oubli illustre de manière emblématique la complexité du problème. Si le bien-fondé de ce droit n'est pas remis en cause, sa mise en œuvre dans la société numérique pose de nombreux problèmes. La décision en mai 2014 de la Cour de justice de l'Union européenne de contraindre Google au déréférencement des pages contenant des informations personnelles à la demande des intéressés, suivant les règles de gestion des données personnelles nationales, soulève d'innombrables problèmes qui ont suscité de vives polémiques.

Elle confère d'abord à Google un pouvoir de plus, laissant la décision de la recevabilité de la demande à un acteur privé, l'administration européenne n'ayant pas les moyens de l'assurer. Elle conduit Google à être informé du souhait des intéressés, enrichissant par là ses connaissances. Elle rend le moteur de recherche un peu plus opaque et limite, dans une perspective très européenne, le droit d'informer. Google, bien sûr hostile à cette décision, a néanmoins déployé les moyens de la mettre en œuvre. Le déréférencement n'étant assuré que sur les domaines européens, la CNIL a demandé en juillet 2015 à Google qu'il soit mis en œuvre sur le domaine `.com` au niveau global, ce que bien sûr Google a refusé. On est au cœur des questions de compétence territoriale qui seront amenées à dominer le débat entre les plateformes et les États dans les années à venir.

Si les États ont une légitimité historique sur leur territoire, les plateformes détiennent désormais un pouvoir croissant qui rentre en conflit direct avec les premiers. Ce pouvoir tient à deux choses. D'abord, au lien direct que les plateformes entretiennent avec les populations et la quantité d'information qu'elles détiennent sur le territoire. Ensuite, au fait que ce lien direct avec les populations permet des intermédiations qui remettent en question des aspects fondamentaux de notre organisation, comme le droit du travail par exemple.

Les données sont une source de pouvoir. Au cours des XIX^e et XX^e siècles, les grandes puissances occidentales ont développé des méthodes de plus en plus subtiles

pour renforcer la connaissance de leur population, avec les recensements dont les questions deviennent progressivement de plus en plus nombreuses et précises, associées aux techniques statistiques, développées tout au long du XIX^e siècle, qui vont accompagner le développement économique et l'augmentation de la qualité de vie.

Désormais les plateformes disposent d'une information sur les populations des territoires qui remet sérieusement en cause les instituts de statistique officiels ainsi que les données émanant d'autres administrations, fiscales, sanitaires, ou autres, dont les sources de données sont à la fois lentes et coûteuses, alors que les plateformes gèrent un flux continu d'information. Si ces données sont de nature différente à ce stade, il est clair que la part relative de données des plateformes augmente au détriment de celle des États.

La répartition géographique des plateformes est non uniforme. Les États-Unis disposent de la majeure partie des plateformes globales, celles qui détiennent une part importante de l'activité nationale dans de nombreux pays. La Chine dispose de près du tiers des grandes plateformes mondiales avec une zone d'influence plutôt régionale aujourd'hui. Quelques petites puissances numériques disposent de plateformes avec une forte activité nationale. L'Europe quant à elle ne dispose d'aucune plateforme importante et dépend fondamentalement des grandes plateformes américaines.

Les conflits entre plateformes et États sont donc pour les pays européens des conflits avec des sociétés étrangères. Si certaines des problématiques sont communes avec celles qu'on trouve aux États-Unis, par exemple sur le droit du travail ou la protection de la vie privée, d'autres, comme nous l'avons vu, s'inscrivent dans une vision du monde – les Européens invoquent parfois leurs valeurs – différente. L'absence d'acteurs de l'intermédiation relève pour l'Europe d'une cruelle défaillance.

Par leur nombre d'utilisateurs, la rapidité de leur pénétration et l'indéniable succès de leurs services, les plateformes d'intermédiation jouissent *de facto* d'une légitimité sur les territoires. Les différentes interdictions, les nombreux procès intentés contre les plateformes, ont permis à ces dernières de sortir renforcées. Il en ira sans doute de même de la décision de justice d'interdire le service Uber Pop en France par exemple. La difficulté est grande pour le législateur de définir le cadre qui permet à ces nouvelles formes d'échanges de se développer dans le respect de règles de fiscalité et d'assurance adaptées à la nouvelle économie.

Les plateformes ont une capacité d'influence très forte. Elles sont en mesure d'orienter le résultat d'une élection et ce sur n'importe quel territoire¹³. De surcroît, les plateformes s'impliquent de manière active dans les votes concernant les nouveaux usages, comme ce fut le cas à San Francisco¹⁴ au mois de novembre 2015,

13. https://www.washingtonpost.com/opinions/could-google-tilt-a-close-election/2013/03/29/c8d7f4e6-9587-11e2-b6f0-a5150a247b6a_story.html

14. http://www.nytimes.com/2015/11/02/technology/san-francisco-ballots-turn-up-anger-over-the-technical-divide.html?emc=edit_tu_20151102&nl=bits&nliid=61924151

à propos d'une proposition de loi sur les locations de courtes durées, que AirBnB contribua à faire échouer. C'est une nouvelle réalité de la vie démocratique, dont les conséquences sont à ce stade difficiles à anticiper. La campagne du président Obama en 2012 s'est appuyée de manière intensive sur les techniques des médias sociaux.

L'histoire des plateformes d'intermédiation remonte essentiellement à une quinzaine d'années. Elle est, par le plus malheureux des hasards, contemporaine des attentats du 11 septembre qui ont fait de la lutte contre le terrorisme une priorité des États. Les tensions actuelles du monde n'infléchiront pas à brève échéance cette tendance. Les services de renseignement, qui ont connu un accroissement rapide de leurs capacités, se sont très rapidement adaptés aux nouvelles réalités des données personnelles pour se rapprocher des nouveaux opérateurs de la donnée, non plus seulement ceux des télécommunications, mais ceux de l'intermédiation dont les données sont d'une qualité bien supérieure. Le débat déclenché par les révélations d'Edward Snowden en juin 2013, qui est loin d'être clos, contribuera aux choix sociétaux du monde en construction.

Être aussi visionnaire que les plateformes dans l'élaboration de nouvelles législations devrait être l'objectif premier des États de l'Europe, mais pour la plupart leurs compétences dans l'économie numérique sont faibles. La bataille entre plateformes et États est inégale. Premièrement, les plateformes sont capables de lever des milliards, leurs valorisations croissent à des vitesses vertigineuses, alors que les États, ceux de l'Europe tout au moins, cherchent à l'inverse à réduire leurs dépenses, laissant des marges de manœuvre tout à fait insuffisantes pour les nouvelles initiatives. Deuxièmement, les plateformes attirent aujourd'hui l'intelligence mondiale. Leur bassin de recrutement est beaucoup plus large que celui de la haute administration des États. Enfin, les plateformes ont une forme d'organisation agile, d'une très grande efficacité, qui contraste avec celle des États, dont la rigidité et la difficulté d'agir sont fortes aujourd'hui.

Conclusion

L'intermédiation algorithmique permet de mettre en relation des acteurs dans un marché biface, sans avoir d'implication dans l'échange. Elle opère au niveau du nuage virtuel, éloigné des nombreuses contraintes du monde physique, ce qui lui permet une agilité très forte. Elle repose sur une connaissance du monde à une granularité inimaginable autrefois, et non disponible aux acteurs traditionnels de l'économie et de l'administration. Elle est donc en capacité de désintermédier les acteurs anciens en réduisant potentiellement à néant leur modèle économique. On peut s'interroger sur l'impact sur nos sociétés, bénéfique ou désastreux. Mais il convient de noter que rien n'arrêtera cette révolution, pas plus qu'on n'a pu stopper la diffusion de l'imprimerie ou celle de l'électricité. Il est clair que l'intermédiation algorithmique redéfinira l'organisation de la cité et de la politique, et que s'il est légitime

de s'en inquiéter, force est de constater que dans le même temps elle offre un formidable espoir pour la gestion des ressources, énergétiques, humaines, en fait de toutes les ressources, qui est sans doute le défi fondamental du millénaire.



L'architecture x86 est-elle sûre ?

Laurent Bloch¹

En octobre 2015 la chercheuse en sécurité informatique Joanna Rutkowska a publié un article retentissant intitulé Intel x86 considered harmful², qui étudie principalement les mécanismes peu connus du démarrage du microprocesseur, avant le lancement du système d'exploitation, et qui fait l'inventaire des failles de sécurité potentielles exploitables par un attaquant.

Le résultat est très impressionnant. Laurent Bloch propose, pour 1024, une analyse précise de cet article.

Lors de la mise sous tension d'un ordinateur il faut que lui soit donnée une information d'amorçage : que faire pour commencer ? Plus précisément : quelle est l'adresse de la première instruction à exécuter ? On imagine bien que la modification de cette adresse par un attaquant pourrait compromettre irrémédiablement tout le fonctionnement ultérieur de quelque logiciel que ce soit, et notamment de tout système d'exploitation, aussi sûr soit-il. Cette question du démarrage est donc cruciale, d'autant plus qu'elle est généralement mal documentée et mal connue. C'est tout le mérite de Joanna Rutkowska d'avoir appliqué ses capacités d'investigation et de critique à ce processus, et d'avoir ainsi montré qu'en dépit d'une complexité considérable ajoutée au cours des dernières années pour en améliorer la sécurité, le système est encore vulnérable. D'où le titre de l'article.

1. lb@laurentbloch.org

2. Joanna Rutkowska. *Intel x86 considered harmful*. The Invisible Things, October 2015, http://blog.invisiblethings.org/papers/2015/x86_harmful.pdf.

Au début était le BIOS

Lorsqu'IBM a lancé les premiers PC en 1981 sur une base de micro-processeurs Intel 8088, il en a publié les spécifications, ce qui a permis que les réalisations ultérieures, éventuellement par d'autres entreprises, respectent une architecture homogène. Selon ces règles architecturales, la carte mère de l'ordinateur reçoit un composant distinct du processeur³ qui contient un programme de démarrage, le *Basic Input Output System* (BIOS). Le circuit de mise sous tension est câblé de façon à lancer l'exécution de la première instruction du BIOS. Le rôle du BIOS consiste à initialiser les composants de la carte mère, à identifier les dispositifs périphériques connectés à l'ordinateur (clavier, écran, souris, disque...), à les mettre en relation avec le processeur. Le BIOS avait été inventé en 1975 par Gary Kildall pour le système CP/M.

L'utilisateur peut accéder au menu de configuration du BIOS pour sélectionner certaines options, par exemple l'ordre selon lequel scruter les périphériques de démarrage possibles pour voir s'ils contiennent un logiciel d'amorçage et un système d'exploitation. Puis le BIOS va charger à partir d'un emplacement convenu sur le périphérique de démarrage choisi (disque dur, clé USB, CD-ROM...) le logiciel d'amorçage (par exemple GRUB dans le cas d'un périphérique de démarrage comportant Linux et éventuellement d'autres systèmes). Le logiciel d'amorçage va permettre à l'utilisateur de choisir le système d'exploitation à lancer s'il y en a plusieurs sur le périphérique de démarrage, et il va le faire démarrer effectivement. Inutile de dire que tout détournement malveillant d'une étape de ce processus peut corrompre toutes les opérations ultérieures, parce qu'à ce stade du démarrage aucun dispositif de protection matériel ou logiciel n'est activé, et qu'ainsi le BIOS a accès sans restriction en mode privilégié à toute la mémoire et à tous les périphériques. Il est par exemple concevable qu'un BIOS corrompu par un attaquant lance une version corrompue du système d'exploitation.

Depuis 2014 la plupart des ordinateurs sont livrés avec un programme de démarrage dit *Unified Extensible Firmware Interface* (UEFI), qui est essentiellement un BIOS écrit selon des principes plus modernes que ses prédécesseurs et doté de fonctions plus étendues, telles que la possibilité de disques de plus grande capacité avec un plus grand nombre de partitions. Un inconvénient majeur d'UEFI est la décision de Microsoft d'imposer aux vendeurs d'ordinateurs certifiés pour Windows 8 ou 10 l'obligation de livrer leur matériel configuré pour démarrer en mode dit *Secure Boot*, contrôlé par une clé de chiffrement privée détenue par Microsoft et utilisée pour signer le noyau du système. Cette situation rend nettement plus difficile l'installation d'un système d'exploitation libre sur ces machines.

3. Historiquement il s'agissait d'un élément de mémoire morte (*Read-Only Memory*, ROM), aujourd'hui il s'agit d'une mémoire flash analogue à celle des clés USB, qui peut être modifiée par logiciel, ce qui est pratique mais introduit un facteur de risque. Le premier virus BIOS, nommé « Chernobyl » ou CIH selon les initiales de son auteur Chen Ing Hau, date du 26 avril 1999.

Confiance et sécurité ?

Joanna Rutkowska nous avertit d'une confusion terminologique : le terme « *trusted* » (de confiance) apposé à un système informatique nous emplit d'un sentiment de sécurité, alors qu'au contraire la crainte devrait nous étreindre. Si celui qui met ce système entre nos mains nous dit qu'il est « de confiance » sans nous donner les moyens de le vérifier, donc en nous contraignant à une confiance *aveugle*, cela signifie que la défaillance de ce système peut détruire *absolument* sa sécurité et tout ce qui en dépend.

Il convient donc de se méfier de ce qui est *trusted*, c'est-à-dire de confiance obli-gée, et d'en réduire le périmètre au minimum. Ce périmètre se nomme « *trusted computing base* » (TCB), il faut en faire sortir le plus possible d'éléments techniques. Comme va nous le montrer Joanna Rutkowska, l'évolution récente de la technolo-gie des processeurs Intel contribue au contraire à élargir démesurément le périmètre technique auquel nous devons faire confiance parce que nous ne disposons d'aucun moyen de savoir comment fonctionne ce qu'il renferme.

Le Bios à la source de toute confiance

Ainsi que nous l'avons signalé ci-dessus, sans confiance dans le BIOS on ne peut plus avoir confiance en rien dans l'ordinateur, son système d'exploitation (OS) et ses données. Joanna Rutkowska précise les raisons de cet état de fait :

- le BIOS est le premier programme à s'exécuter sur le processeur, ce qui lui permet de corrompre l'image de l'OS ou de l'hyperviseur chargée ultérieurement ;
- le BIOS a accès en mode privilégié à l'ensemble du matériel, ce qui lui permet de programmer des actions malfaisantes (par exemple les attaques dites DMA et *Evil Maid*) ;
- le BIOS fournit le code qui s'exécute selon le *System Management Mode* (SMM) ; or le code SMM peut s'exécuter pendant toute la période d'activité du système, d'où un risque majeur induit par sa corruption éventuelle ; Xeno Kovah et Corey Kallenberg ont prouvé la réalité de ce risque en écrivant le *rootkit* SMM *LightEater*⁴ capable de voler des clés privées en mémoire vive pendant l'exécution du système *Tails OS*⁵, pourtant spécialement conçu à partir d'un système Linux Debian pour la sécurité (ses communications réseau utilisent Tor, Edward Snowden l'utilise pour cette raison).

4. Xeno Kovah, Corey Kallenberg, *How Many Million BIOSes Would you Like to Infect?* LegbaCore, June 2015, http://legbacore.com/Research_files/HowManyMillionBIOSesWouldYouLikeToInfect_Whitepaper_v1.pdf.

5. *Tails : The amnesic incognito live system*. The TAILS Project, November 2015, <https://tails.boum.org/>.

Le BIOS est donc un élément ultra-sensible de tout système informatique à base de processeur x86, or comme nous l'avons écrit plus haut il est modifiable assez facilement, l'article de Joanna Rutkowska indique plusieurs procédés pour ce faire, et il n'existe à l'heure actuelle aucun moyen de s'affranchir de cette dépendance vis-à-vis du BIOS, en d'autres termes de faire en sorte qu'il puisse être considéré comme non-fiable sans que ce soit grave.

Au-delà de cette analyse des pouvoirs du BIOS, Joanna Rutkowska commence à nous faire prendre conscience de la présence d'une quantité considérable de logiciel actif pendant toute la vie du système et hors de tout contrôle du système d'exploitation. Il serait donc vital que ce logiciel soit vérifiable et sûr. Nous allons voir qu'il n'en est rien.

Multiples sous-systèmes en micro-code

Pour corriger certaines failles de sécurité telles que celles mentionnées ci-dessus, les ingénieurs d'Intel ont multiplié les dispositifs : *Trusted Platform Module* (TPM), *Trusted Execution Technology* (TXT), exécution du code SMM (*System Management Mode*) sous contrôle d'un hyperviseur, *Active Management Technology* (AMT), *Boot Guard*, *Secure Boot* pour UEFI, et pour couronner l'édifice *Intel Management Engine* (ME). Il s'agit en fait de systèmes implantés dans le micro-code du processeur, c'est-à-dire hors de portée de l'utilisateur même muni de tous les privilèges sur son système. L'article de notre auteure en fait une analyse approfondie.

Ce qui est à noter, c'est que des systèmes comme AMT et ME sont actifs en permanence, *y compris lorsque l'ordinateur est arrêté* ! C'est dire à quel niveau de contrôle arrivent ces techniques, dont la puissance est telle qu'il est à souhaiter qu'elles ne puissent pas être détournées par des malfaisants.

Intel Management Engine (ME)

Nous ne traiterons pas ici de tous ces dispositifs, pour lesquels nous renvoyons à l'article, et nous n'évoquerons que ME (*Intel Management Engine*), parce que c'est le dernier en date et qu'il englobe en quelque sorte tous les autres. ME est un petit ordinateur incorporé à *tous* les processeurs Intel contemporains. Il est impossible de l'enlever ou de le désactiver, et d'ailleurs, à supposer que l'on trouve un moyen de l'inhiber, le processeur deviendrait pratiquement inutilisable parce que beaucoup de ses fonctions dépendent de ME.

Plus qu'un simple microcontrôleur, ME est une infrastructure d'accueil complète pour toutes sortes de sous-systèmes, et de fait AMT est désormais implanté sur ME, ainsi que *Boot Guard*, PTT (*Platform Trust Technology*, la version pour ME de TPM), et d'autres à venir. ME procure bien sûr un hyperviseur qui permet par exemple l'exécution de code SMM en bac à sable (pour une récapitulation de tous

ces sigles on pourra se reporter au document *Intel Hardware-based Security Technologies for Intelligent Retail Devices*⁶).

ME partout et pour tout ?

Joanna Rutkowska souligne les effets pervers de cette prise de contrôle par ME de tous les aspects du fonctionnement du système.

D'abord, imposer à tous les utilisateurs de processeurs Intel une technologie fermée et opaque en prétendant qu'elle offrirait un niveau de sécurité inégalable, sans discussion possible, est une attitude arrogante et peu convaincante sur le fond.

Ensuite, si cette idée s'impose, et il semble que l'on n'ait guère le choix (AMD développe des technologies comparables sous le nom *Platform Security Processor*, PSP), cela conduira à ce qu'elle appelle la « zombification » des systèmes d'exploitation tels que Windows, Linux, OS-X, etc., réduits au rôle d'interfaces avec l'utilisateur pour balader des fenêtres, réagir aux déplacements de souris et jouer de la musique, cependant que les véritables traitements de données seront effectués derrière les portes closes (par des clés de chiffrement détenues par Intel) de ME, sans que l'utilisateur sache quels sont les algorithmes utilisés, et avec quel niveau de sécurité.

Comment savoir, par exemple, si Intel n'a pas décidé de confier le séquestre de ses clés de chiffrement à un tiers de confiance, par exemple une agence de sécurité gouvernementale américaine ? ME pourra-t-il filtrer et analyser nos messages électroniques, nos communications par Skype, nos recherches sur le Web ? Le générateur de nombres pseudo-aléatoires de ME comporte-t-il des faiblesses, voulues ou non ? Répondre à de telles questions est déjà difficile aujourd'hui, mais Joanna Rutkowska attire notre attention sur le fait qu'avec ME la rétro-ingénierie nécessaire à leur compréhension et à leur analyse sera d'une difficulté accrue d'un ordre de grandeur.

Et si ME était corrompu ?

C'est la question qu'il faut toujours se poser à propos d'un système tout-puissant : tant que ses actions sont bénéfiques tout va bien⁷, mais s'il est détourné vers des actions maléfiques, intentionnellement ou par suite d'une attaque réussie, alors l'empire du mal risque d'être absolu.

Or, pour qui voudrait entreprendre une action maléfique, ME est l'infrastructure idéale, nous dit Joanna Rutkowska : un *rootkit* implanté dans ME aurait le contrôle total des traitements effectués par le système et des données traitées (y compris les clés privées en transit par la mémoire), et il serait pratiquement indétectable. Que ce *rootkit* soit implanté par la mafia ou par la NSA ne change rien au problème.

6. <http://www.intel.com/content/www/us/en/intelligent-systems/shark-bay/security-technologies-4th-gen-core-retail-paper.html>

7. Sous réserve d'un accord unanime pour déterminer ce qui est bénéfique et ce qui ne l'est pas, et l'on sait que ce n'est vrai que dans l'imagination des dictateurs totalitaires, c'est même la définition du totalitarisme.

Idées pour un système plus sûr

La démarche de Joanna Rutkowska lui permet d'élaborer les idées qui mènent à un système plus sûr, par exemple :

- utilisation intensive des techniques d'isolation des différents artefacts fonctionnels les uns par rapport aux autres : virtualisation et bac à sable (*sandboxing*) en particulier ;
- exécuter en mode non-privilegié tout ce qui peut l'être, et son article démontre que c'est possible pour presque tout ce qui a trait aux périphériques, ce qui annulerait les menaces de type *Evil Maid* et DMA par exemple.

Joanna Rutkowska ne s'est pas contentée de proférer des idées, qu'elle réunit sous le vocable compartimentation, elle les a mises en pratique en organisant le laboratoire *The Invisible Things* qui a créé le système d'exploitation Qubes OS⁸ fondé sur ces principes. On remarquera que cette idée de décomposer les systèmes en sous-systèmes le plus possible indépendants les uns des autres et dotés de privilèges réglés au minimum rejoint le courant des micro-noyaux, une technologie un peu oubliée mais dont les qualités de modularité, de flexibilité et d'aptitude au calcul réparti devraient permettre le retour au premier plan.

Elle constate avec dépit que de façon générale les industriels et les éditeurs de systèmes d'exploitation ne suivent pas ces principes et continuent à produire des systèmes monolithiques et donc vulnérables.

Il n'en reste pas moins que même en supposant toutes ces idées mises en œuvre, la question du démarrage reste entière, parce qu'elle est entre les mains des industriels qui conçoivent et fabriquent processeurs et cartes mères. La multiplication par Intel de technologies concurrentes ou complémentaires ne fait que compliquer la question : *Trusted Platform Module* (TPM), *Trusted Execution Technology* (TXT), exécution du code SMM (*System Management Mode*) sous contrôle d'un hyperviseur, *Boot Guard*, *Secure Boot* pour UEFI, et pour couronner l'édifice *Intel Management Engine* (ME), qui est incorporé de façon irréversible à tous les processeurs Intel contemporains et qui est un véritable système d'exploitation implanté en micro-code et qui vit sous le système d'exploitation que l'utilisateur croit avoir choisi.

Avec *Intel Management Engine* le vrai système d'exploitation est dans les entrailles du processeur, et Windows, Linux ou OS-X ne sont plus que des systèmes de gestion de fenêtres, d'affichage de vidéo et de diffusion de musique. Ce qui est grave dans tout cela, c'est que non seulement l'utilisateur (et même le fabricant d'ordinateur) est privé de toute liberté de choix de son système, mais qu'en outre le système imposé n'offre pas et ne peut pas offrir les garanties de sécurité auxquelles il prétend.

8. <https://www.qubes-os.org/>



Des consensus tout sauf mous

Anne-Marie Kermarrec¹

Michel Raynal, Professeur à l'université de Rennes 1 et membre senior de l'Institut universitaire de France est un passionné. Il aime les voyages, adore la littérature, ne se lasse pas des côtes bretonnes (malgré ses origines du Sud-Ouest), encore moins du rugby (probablement en raison de ces mêmes origines). Michel aurait pu devenir un grand scientifique dans bien des domaines. C'est dans l'informatique qu'il est tombé, et plus particulièrement dans cette même région de l'Ouest où d'autres se sont retrouvés dans une marmite de potion magique. Michel Raynal est le lauréat du SIROCCO Innovation Award 2015 pour ses travaux en informatique distribuée. C'est l'occasion de découvrir ce domaine passionnant.

Un *système distribué* est un système composé de plusieurs entités (téléphone, capteur, ordinateur par exemple), connectées par un réseau de communication, c'est-à-dire qui peuvent communiquer (en filaire, par wifi, etc.) et qui, ensemble, s'attaquent à un problème comme de réaliser un calcul ou chercher de l'information. Nous vivons aujourd'hui dans un monde où la majorité des systèmes est distribuée. Par exemple, les données relatives aux comptes Facebook d'un ensemble d'amis sont dispersées aux quatre coins du monde ; pourtant lorsqu'un utilisateur met à jour son statut, tous ses amis doivent être notifiés, qu'ils soient connectés depuis le Maroc ou la Chine et que leurs données soient hébergées en Inde ou dans la Silicon Valley. Nos données, comme celles des entreprises, sont distribuées sur des serveurs ; on parle du *cloud*. La même donnée peut être répliquée sur plusieurs serveurs pour des raisons de fiabilité notamment.

1. Inria Rennes, Anne-Marie.Kermarrec@inria.fr.

Turing s'est un jour posé la question de ce que sa machine universelle était capable de calculer. Malheureusement, son formalisme ne s'applique pas aux systèmes distribués. Évidemment, les experts se sont posé la question de savoir ce qu'il est possible de calculer dans un système distribué, notamment le problème du consensus qui a été particulièrement étudié. Le problème est le suivant : chaque entité propose une valeur et, à la fin du calcul, toutes doivent s'être mises d'accord sur l'une de ces valeurs. C'est le type de fonctionnalité dont on a besoin, par exemple, pour assurer la cohérence de données répliquées dans le *cloud*. Pour illustrer le besoin de consensus, prenons l'exemple de données bancaires répliquées sur les machines A en France et B aux États-Unis. Ces données sont répliquées pour des raisons de tolérance aux pannes, de sorte que si une machine contenant ces données tombe en panne, l'autre continue de fonctionner. Ceci permet en outre aussi aux utilisateurs des États-Unis d'avoir un accès rapide depuis la machine B et aux utilisateurs européens d'avoir un accès rapide depuis la machine A. Considérons maintenant la situation suivante : Alice effectue un débit de 100 \$ depuis la France (sur la machine A) de son compte, dont le solde est de 1000 \$. L'ordre de débit est envoyé à la machine A mais aussi à la machine B pour assurer que les comptes soient cohérents. Le banquier lui, depuis la machine B, décide de verser des intérêts de 10% sur ce compte. Cet ordre de calcul d'intérêt est aussi envoyé aux deux machines. La communication prend du temps pour traverser l'Atlantique, si bien que la machine A reçoit d'abord l'ordre de débit puis celui des intérêts, appliquant donc 10% à un solde de 900 \$, le solde résultant est alors de 990 \$, alors que la machine B, applique les intérêts, puis le débit, le solde étant alors de 1000 \$. Pour assurer un bon fonctionnement il est essentiel que les deux machines exécutent les instructions dans le même ordre et, pour ce faire, elles doivent atteindre un consensus.

Fisher, Lynch et Patterson [1] ont montré en 1985 que le consensus était impossible dans un système réparti asynchrone² dans lequel les machines peuvent tomber en panne. La preuve est longue et intriquée et il faudrait demander à Michel Raynal de nous l'expliquer. Pour faire court, dans un système asynchrone, si une entité ne répond pas, on ne sait pas différencier si c'est une question de lenteur, ou si c'est une défaillance de l'entité. Et on peut soit attendre éternellement le message alors que l'entité est défaillante ou alors prendre une décision et voir le message arriver trop tard et contredire notre décision. Michel Raynal a reçu le *Sirocco Innovation Award* en particulier pour ses travaux relatifs à ce problème de consensus, développant de nouveaux résultats à partir du théorème d'impossibilité de Fisher, Lynch et Patterson.

2. Dans un système synchrone, les opérations sont coordonnées sous un contrôle centralisé basé sur les signaux d'une horloge. Par opposition, un système asynchrone, en revanche, n'a pas d'horloge globale. Les différentes entités doivent utiliser des communications pour coordonner leurs tâches.

Que peut faire un chercheur en algorithmique distribuée quand un problème est impossible à résoudre ? Il étudie les hypothèses à relâcher pour que le problème devienne résoluble. Par exemple, pour le problème du consensus, on peut décider de prendre l'hypothèse que les messages entre les entités prennent un temps borné. Le consensus devient alors possible. Que peut-il faire quand on sait résoudre le problème ? Il peut chercher un algorithme plus efficace, par exemple plus rapide, demandant moins de messages, moins de calcul. Afin de contourner l'impossibilité du consensus en asynchrone, Michel Raynal, avec ses collègues Achour Mostéfaoui et Sergio Rajsbaum, a par exemple proposé une nouvelle méthode à base de conditions [2]. L'idée est de dénombrer les « conditions », c'est-à-dire les configurations pour lesquelles on sait comment obtenir un consensus. Michel a alors conçu des algorithmes qui permettaient de s'adapter à ces conditions, rendant ainsi l'obtention d'un consensus possible [3].

Le prix de Michel Raynal reconnaît ses contributions scientifiques. Pour conclure, on pourrait souligner que Michel assure la relève en formant des jeunes chercheurs, et en leur transmettant son enthousiasme et sa passion.

Références

- [1] Michael J. Fisher, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the Association for Computing Machinery* 32(2):374–382 (1985).
- [2] Achour Mostéfaoui, Sergio Rajsbaum and Michel Raynal. Conditions on input vectors for consensus solvability in asynchronous distributed systems. *Journal of the Association for Computing Machinery* 50(6):922–954 (2003).
- [3] Achour Mostéfaoui, Sergio Rajsbaum and Michel Raynal. Efficient Condition-Based Consensus. SIROCCO 2001.

Cet article est également paru sur le blog BINAIRE³.

3. <http://binaire.blog.lemonde.fr/>



Michel Raynal élu membre de *Academia Europaea*

Michel Raynal, membre senior de l'Institut universitaire de France, professeur à l'IRISA, (Université de Rennes 1), membre du Conseil d'administration et du Conseil scientifique de la SIF, vient d'être élu membre de *Academia Europaea*¹.

Il est le dix-huitième informaticien français ainsi honoré au niveau européen, et rejoint ainsi Serge Abiteboul, Jean-Raymond Abrial, Jacques Arzac, Gérard Berry, Marie-Paule Cani, Giuseppe Castagna, Patrick Cousot, Erol Gelenbe, Jean-Yves Girard, Nicolas Halbwachs, Gérard Huet, Anne-Marie Ker-marrec, Giuseppe Longo, Maurice Nivat, Dominique Perrin, Jean-Eric Pin et Joseph Sifakis.



Academia Europaea est une académie européenne, créée en 1988, dont le but est de promouvoir l'éducation et la recherche. Elle comprend actuellement près de 4000 membres (dont soixante prix Nobel) issus de trente-cinq pays. Du point de vue

1. Plus d'informations sur cette institution peuvent être trouvées via les liens suivants : <http://www.ae-info.org/> (en anglais), ou https://fr.wikipedia.org/wiki/Academia_Europaea (en français).

thématique, *Academia Europaea* inclut – entre autres – les domaines suivants : mathématiques, informatique, sciences sociales, médecine, physique, chimie, biologie, économie, droit, lettres et humanités, qui forment autant de départements de l'académie.

La SIF se réjouit de cette nomination et adresse à Michel Raynal ses plus chaleureuses félicitations.



Cinquante boîtes fermées

Jean-Paul Delahaye¹

La rubrique « Récréation informatique » propose une petite énigme algorithmique ou sur un thème de mathématiques discrètes susceptible d'intéresser un lecteur de 1024. La solution est donnée dans le numéro suivant.

Rappel et solution du problème précédent

MOINS DE 20 CARTES

Cent nombres différents sont écrits sur cent cartes différentes, qu'on dispose à plat sur une table en 10 lignes de 10 cartes. Dans chaque ligne et dans chaque colonne, les nombres sont classés par ordre croissant (de gauche à droite et de haut en bas). Les cartes ont été tournées et vous ne voyez donc pas les nombres qui y sont inscrits. Trouvez un algorithme qui, pour tout nombre donné, vous indique si oui ou non il est présent sur une des cent cartes, et cela en retournant moins de 20 cartes.

SOLUTION. Merci aux lecteurs qui m'ont fait parvenir des bonnes réponses. Ce sont dans l'ordre d'arrivée des messages : Arnaud Casteigts et Jérémie Decouchant.

On commence en retournant la carte en haut à droite. On en compare la valeur avec celle recherchée. Si c'est la bonne valeur, c'est terminé. Si la valeur recherchée est plus petite que celle de la carte retournée, la carte recherchée ne peut pas être dans la dernière colonne : on se déplace sur la carte à gauche de celle qu'on vient de retourner. Si la valeur recherchée est plus grande que celle de la carte retournée, la carte recherchée ne peut pas être dans la première ligne : on se déplace sur la carte

1. Université de Lille 1, Sciences et Technologies, Centre de recherche en informatique signal et automatique de Lille (CRISAL), UMR 9189 CNRS, Bât M3-ext, 59655 Villeneuve d'Ascq Cedex. E-mail : jean-paul.delahaye@univ-lille1.fr.

juste en dessous de celle qu'on vient de retourner. On recommence alors l'opération. À chaque étape, on se déplace d'une case vers la gauche ou d'une case vers le bas, et tout ce qui se trouve à droite de la case active ou au-dessus ne peut contenir la carte recherchée. La plus longue série de cartes retournées quand on poursuit ce chemin en zigzag vers le bas à gauche en comporte 19. En retournant 19 cartes au plus, on est donc assuré, soit de trouver la carte recherchée si elle est présente, soit d'avoir la certitude qu'elle n'est pas parmi les cartes sur la table.

Nouveau Problème

CINQUANTE BOÎTES FERMÉES

Voici un problème extraordinaire tant il paraît impossible à résoudre.

Cinquante prisonniers ont l'opportunité d'être libérés tous ensemble s'ils réussissent la difficile épreuve suivante. Les gardiens de la prison écrivent les cinquante noms des prisonniers sur cinquante cartons placés dans cinquante boîtes fermées et alignées sur une grande table. Sans savoir ce qu'auront fait les prisonniers précédents, chaque prisonnier est conduit dans la salle où se trouve la table et doit ouvrir 25 des cinquante boîtes et y trouver son nom. L'épreuve n'est réussie que si chaque prisonnier trouve son nom. Avant que l'épreuve commence, les prisonniers peuvent échanger entre eux pour convenir d'une méthode, mais une fois l'épreuve commencée, ils n'ont plus aucun échange entre eux. Si chaque prisonnier procédait au hasard leur chance de réussir serait de

$$\frac{1}{250} = 8,881 \times 10^{-6},$$

ce qui est très très peu !

Ils peuvent faire beaucoup mieux et avoir une probabilité de réussite supérieure à 30 %. Comment ?

Envoyez vos réponses à jean-paul.delahaye@univ-lille1.fr. Le nom des premiers lecteurs à me donner la bonne réponse (et à la justifier) seront mentionnés dans le prochain numéro de 1024.



B U L L E T I N

de la société informatique
de France



Institut Henri Poincaré,
11 rue Pierre et Marie Curie,
75231 Paris Cedex